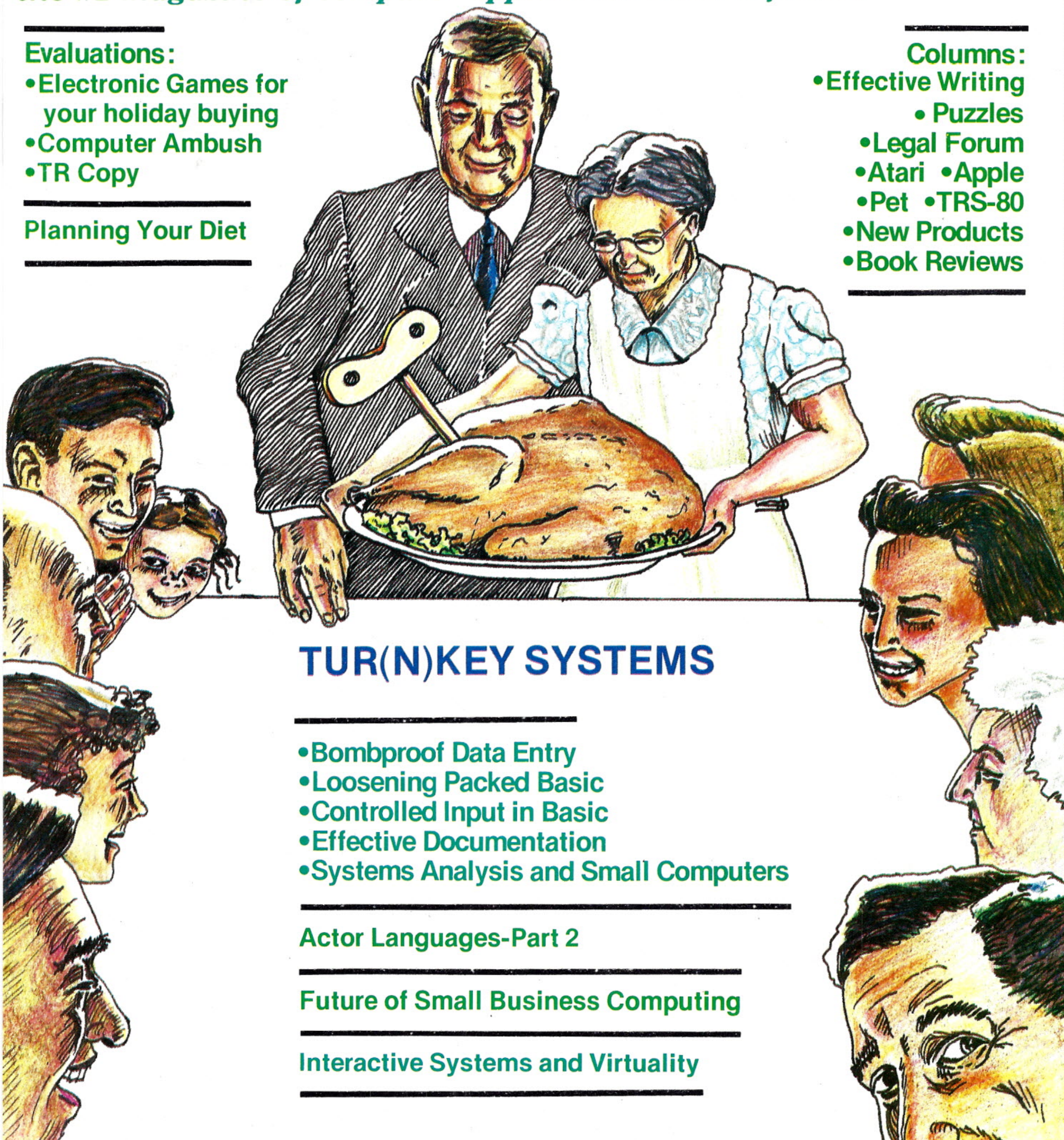
November 1980
vol 6 [illegible] 11
$3.00

creative computing

the #1 magazine of computer applications and software



Evaluations:
- Electronic Games for your holiday buying
- Computer Ambush
- TR Copy

Planning Your Diet

Columns:
- Effective Writing
- Puzzles
- Legal Forum
- Atari •Apple
- Pet •TRS-80
- New Products
- Book Reviews

TUR(N)KEY SYSTEMS

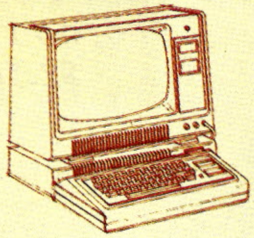
- Bombproof Data Entry
- Loosening Packed Basic
- Controlled Input in Basic
- Effective Documentation
- Systems Analysis and Small Computers

Actor Languages-Part 2

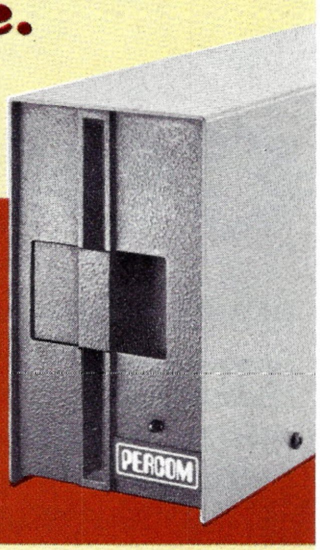
Future of Small Business Computing

Interactive Systems and Virtuality

TRS-80* Model I Computer Owners . . .



Double-density storage. It's really here!



Here at Percom. And your authorized Percom dealers.

And double-density storage is here in a big way. Because now you can choose from *three different levels* of mini-disk systems — all *double-density rated.*

And get the storage that precisely meets your application needs.

Not to mention the service and quality that's made Percom the industry leader.

Although rated for double-density operation, all levels of Percom drives *work equally well in single-density* applications.

You can operate these drives in ordinary single-density format using TRSDOS*, Percom OS-80™ or any other single-density operating system.

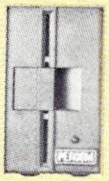
Or, you can add a Percom DOUBLER™ to your Tandy Expansion Interface and store data and programs in *either* single- or double-density format.

Under double-density operation, you can store as much as *350 Kbytes* of formatted data — depending on the drive model — on one side of a five-inch minidiskette.

That's *four times* the capacity of standard Model I mini-disks, almost 100 Kbytes *more than* the capacity of the *eight-inch* IBM 3740 format!

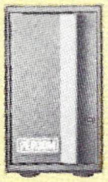
Available in 1-, 2- and 3-drive configurations in all three model lines, Percom *burned-in, fully-tested* drives start at only $399.

TFD-40™ Drives



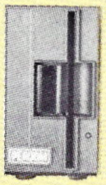
TFD-40 Drives store 180 Kbytes (double-density) or 102 Kbytes (single-density) of **formatted** data on one side of a 40-track minidiskette. Although economically priced, TFD-40 drives receive the same full Percom quality control measures as TFD-100 and TFD-200 drives.

TFD-100™ Drives



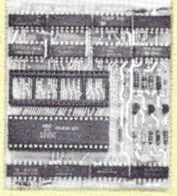
TFD-100 drives are "flippy" drives. You store twice the data per minidiskette by using both sides of the disk. TFD-100 drives store 180 Kbytes (double-density) or 102 Kbytes (single-density) **per side.** Under double-density operation, you can store a 70-page document on one minidiskette.

TFD-200™ Drives



TFD-200 drives store 350 Kbytes (double-density) or 197 Kbytes (single-density) on one side of a minidiskette. By comparison, 3740-formatted eight-inch disks store only 256 Kbytes. Enormous on-line storage capacity in a 5″ drive, plus proven Percom reliability. That's what you get in a TFD-200.

the DOUBLER™ — This proprietary adapter for the TRS-80* Model I computer packs approximately twice the data on a disk track.



Depending on the type of drive, you can store up to four times as much data — 350 Kbytes — on one side of a minidiskette as you can store using a Tandy standard Model I computer drive.

Easy to install, the DOUBLER merely plugs into the disk controller chip socket of your Expansion Interface. No rewiring. No trace cutting.

And because the DOUBLER reads, writes and formats *either* single- or double-density disks, you can continue to run all of your single-density software, then switch to double-density operation at any convenient time.

Included with the PC card adapter is a TRSDOS*-compatible double-density disk operating system, called DBLDOS™, plus a CONVERT utility that converts files and programs from single- to double-density or double- to single-density format.

Each DOUBLER also includes an on-card high-performance *data separator circuit* which ensures reliable disk read operation.

The DOUBLER works with standard 35-, 40-, 77- and 80-track drives rated for double-density operation.

Note. Opening the Expansion Interface to install the DOUBLER may void Tandy's limited 90-day warranty.

Drive enclosures, power supplies Percom drive enclosures are finished in compatible silver enamel. Three sizes accommodate either 1, 2 or 3 drives. Drive power supplies are heavy duty, cool-running open-frame design. Three-wire ac power cords are safer, have lower noise pickup.

Free software patch This software patch, called PATCH PAK™, upgrades TRSDOS* for operation with improved 40- and 77-track drives. For single-density operation only.

Quality Percom products are available at authorized dealers. Call *toll free 1-800-527-1592* for the address of your nearest dealer or to order directly from Percom. Prices and specifications subject to change without notice.

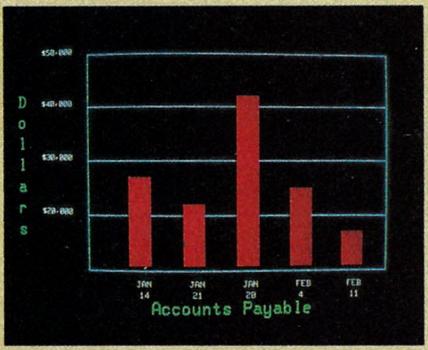
™ trademark of Percom Data Company, Inc.

* trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.

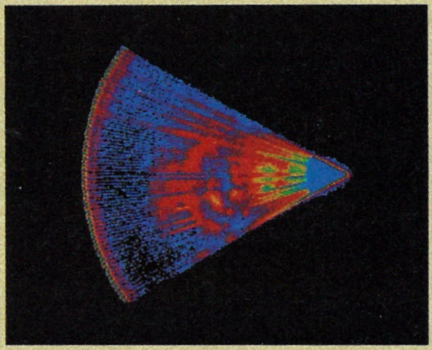
PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY • GARLAND TX • 75042
(214) 272-3421

CIRCLE 177 ON READER SERVICE CARD



Management Information Display



Ultrasonic heart sector scan

High-resolution display with alphanumerics

Get the professional color display that has BASIC/FORTRAN simplicity

LOW-PRICED, TOO

Here's a color display that has everything: professional-level resolution, enormous color range, easy software, NTSC conformance, and low price.

Basically, this new Cromemco Model SDI* is a two-board interface that plugs into any Cromemco computer.

The SDI then maps computer display memory content onto a convenient color monitor to give high-quality, high-resolution displays (756 H x 482 V pixels).

When we say the SDI results in a high-quality professional display, we mean **you can't get higher resolution than this system offers in an** NTSC-conforming display.

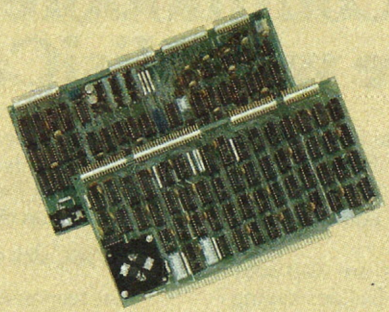
The resolution surpasses that of a color TV picture.

BASIC/FORTRAN programming

Besides its high resolution and low price, the new SDI lets you control with optional Cromemco software packages that use simple BASIC- and FORTRAN-like commands.

Pick any of 16 colors (from a 4096-color palette) with instructions like DEFCLR (c, R, G, B). Or obtain a circle of specified size, location, and color with XCIRC (x, y, r, c).

*U.S. Pat. No. 4121283



Model SDI High-Resolution Color Graphics Interface

HIGH RESOLUTION

The SDI's high resolution gives a professional-quality display that strictly meets NTSC requirements. You get 756 pixels on every visible line of the NTSC standard display of 482 image lines. Vertical line spacing is 1 pixel.

To achieve the high-quality display, a separate output signal is produced for each of the three component colors (red, green, blue). This yields a sharper image than is possible using an NTSC-composite video signal and color TV set. Full image quality is readily realized with our high-quality RGB Monitor or any conventional red/green/blue monitor common in TV work.

Model SDI plugs into Z-2H 11-megabyte hard disk computer or any Cromemco computer

DISPLAY MEMORY

Along with the SDI we also offer an optional fast and novel **two-port** memory that gives independent high-speed access to the computer memory. The two-port memory stores one full display, permitting fast computer operation even during display.

CONTACT YOUR REP NOW

The Model SDI has been used in scientific work, engineering, business, TV, color graphics, and other areas. It's a good example of how Cromemco keeps computers in the field up to date, since it turns any Cromemco computer into an up-to-date color display computer.

The SDI has still more features that you should be informed about. So contact your Cromemco representative now and see all that the SDI will do for you.

Cromemco
incorporated

280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400

Tomorrow's computers today

CIRCLE 130 ON READER SERVICE CARD

In This Issue

evaluations & profiles

articles

fiction & foolishness

NOVEMBER 1980 **VOLUME 6, NUMBER 11**

Creative Computing magazine is published monthly by Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. (Editorial office 51 Dumont Place, Morristown, NJ 07960 Phone: (201) 540-0445.)

Domestic Subscriptions: 12 issues $15,24 issues $28,36 issues $40. Send subscription orders or change of address (P.O. Form 3575) to Creative Computing P.O. Box 789-M, Morristown, NJ 07960. Call 800-631-8112 toll-free (in New Jersey call 201-540-0445) to order a subscription (to be charged only to a bank card).

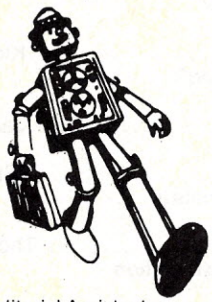
Controlled Circulation pending at Richmond, VA 23228.

applications - games

departments

The cover is by Dan Smith of Swarthmore, Pennsylvania. Six articles in this issue deal with making systems easy to use (or almost turnkey).

Publisher/Editor-in-chief	**David H. Ahl**
Editor	**Ted Nelson**
Managing Editor	**Burchenal Green**
Associate Editor	**David Lubar**
Contributing Editors	**George Blank** **Frederick Chesson** **Charles Carpenter** **Margot Critchfield** **Thomas W. Dwyer** **Stephen B. Gray** **Richard Kaapke** **Stephen Kimmel** **Harold Novick** **Peter Payack** **Alvin Toffler** **C. Barry Townsend** **Gregory Yob** **Karl Zinn**
Editorial Assistant	**Peter Fee**
Secretary	**Elizabeth Magin**
Production Manger	**A.J. Siciliano**
Art Department	**Deborah Rommel** **Diana Negri** **Chris DeMilia** **Patricia Balinski**
Typesetter	**Debbie Barbagallo**
Advertising Sales	**Renee Fox Christman** **Nancy Wood**
Marketing Coordinators	**Barbara Garris** **David Rogers**
Software Development	**Laura McLaughlin** **Chris Vogeli** **Bob Callan**
Systems Analyst	**David Gewirtz**
Data Processing Coordinator	**Keith Franklin**
Software Production	**Debra Linton** **Rita Gerner**
Business Manager	**Elizabeth Staples**
Financial Coordinator	**William L. Baumann**
Bookkeeper	**Patricia Kennelly**
Retail Marketing	**Jennifer Burr** **Laura Gibbons**
Circulation	**Suzanne Guppy** **Frances Miskovich** **Moira Fenton** **Carol Vita** **Dorothy Staples**
Customer Service	**Patricia Brown**
Office Assistants	**Rosemary Bender** **Linda McCatharn** **Maria Petrakis**
Order Processing	**Jim Zecchin** **Alan Kelly** **Dorian Snipes** **Joan Swihart**
Book Service Supervisors	**Ronald Antonaccio** **William Rogalski**
Book Service	**Scott McLeod** **Nick Ninni** **Mark Archambault** **Mike Gribbon**



Advertising Sales

Advertising Coordinator
Renee Christman
Creative Computing
P.O. Box 789-M
Morristown, NJ 07960
(201)540-9168

Western State, Texas
Jules E. Thompson, Inc.
1290 Howard Ave., Suite 303
Burlingame, CA 94010
(415)348-8222

Southern California
Jules E. Thompson, Inc.
2560 Via Tejon
Palos Verdes Estates, CA 90274
(213)378-8361

Mid-Atlantic, Northeast
CEL Associates, Inc.
36 Sohier Street
Cohasset, MA 02025
(617)383-6136

Midwest
Ted Rickard
435 Locust Rd.
Wilmette, IL 60091
(312)251-2541

New York Metropolitan Area
Nelson & Miller Associates, Inc.
55 Scenic Dr.
Hastings-on-Hudson, NY 10706
(914)478-0491

Southeast
Paul McGinnis Co.
60 East 42nd St.
New York, NY 10017
(212)490-1021

Responsibility

Creative Computing will not be responsible for the return of unsolicited manuscripts, cassettes, floppy disks, program listings, etc. not submitted with a self-addressed, stamped envelope.

OK to Reprint

Material in Creative Computing may be reprinted without permission by school and college publications, personal computing club newsletters, and non-profit publications. Only original material may be reprinted; that is, you may not reprint a reprint. Also, each reprint must carry the following notice on the first page of the reprint in 7-point or larger type (you may cut out and use this notice if you wish):

Please send us two copies of any publication that carries reprinted material. Send to attention: David Ahl.

Microform

Creative Computing is available on permanent record microfilm. For complete information contact University Microfilms International, Dept. F.A., 300 North Zeeb Road, Ann Arbor, MI 48106 or 18 Bedford Road, London WC1R 4EJ, England.

Foreign Customers

Foreign subscribers in countries listed below may elect to subscribe with our local agents using local currency. Of course, subscriptions may also be entered directly to Creative Computing (USA) in U.S. dollars. (bank draft or American Express card). All foreign subscriptions must be prepaid.

Many foreign agents stock Creative Computing magazines, books, and software. However, please inquire directly to the agent before placing an order. Again, all Creative Computing products may be ordered direct from the USA — be sure to allow for foreign shipping and handling.

CANADA	**Surface**	**Air**
1-year	C $28	n/a
2-year	54	
3-year	78	

Micron Distrib.
409 Queen St. W.
Toronto, OT M5V 2A5, Canada

ENGLAND	£	£
1-year	10.00	19.00
2-year	19.50	37.25
3-year	28.50	55.00

CREATIVE COMPUTING
Attn: Hazel Gordon
27 Andrew Close
Stoke Golding, Nuneaton CV13 6EL
England

FRANCE	F	F
1-year	98	183
2-year	188	358
3-year	273	530

SYBEX EUROPE
14/18 Rue Planchat
75020 Paris, France

SWEDEN	kr	kr
1-year	100	188
2-year	193	368
3-year	280	544

HOBBY DATA
Attn: Jan Nilsson
Fack
S-200 12 Malmo 2, Sweden

GERMANY	dm	dm
1-year	42	78
2-year	80	152
3-year	116	225

HOFACKER-VERLAG
Ing. W. Hofacker
8 Munchen 75
Postfach 437, West Germany

HOLLAND, BELGIUM		f
1-year		108
2-year		210
3-year		300

2XF COMPUTERCOLLECTIEF
Attn: F. de Vreeze
Amstel 312A
1017 AP AMSTERDAM, Holland

AUSTRALIA	$A	$A
1-year	23	47
2-year	44	92
3-year	64	136

ELECTRONIC CONCEPTS PTY., LTD.
Attn: Rudi Hoess
Ground Floor 55 Clarence St.
Sydney, NSW 2000, Australia

JAPAN	Y	Y
1-year	5,700	10,700
2-year	10,900	21,000
3-year	15,900	31,000

ASCII PUBLISHING
Aoyama Building 5F
5-16-1 Minami Aoyama, Minato-Ku
Tokyo 107, Japan

HONG KONG	$HK	$HK
1-year	118	222
2-year	227	435
3-year	330	640

COMPUTER PUBLICATIONS, LTD.
22 Wyndham St., 7th Floor
Hong Kong

PHILIPPINES	P	P
1-year	175	330
2-year	338	650
3-year	490	955

INTEGRATED COMPUTER SYSTEMS, INC.
Suite 205, Limketkai Bldg., Ortigas Ave.
Greenhills P.O. Box 483, San Juan
Metro Manila 3113, Philippines

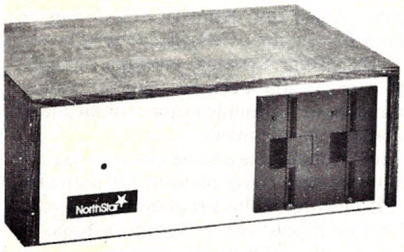
OTHER COUNTRIES		
1-year	$24	$45
2-year	46	88
3-year	67	130

CREATIVE COMPUTING
P.O. Box 789-M
Morristown, N.J. 07960, USA

If North Star or Cromemco offer it . . .

WE HAVE IT!!

Immediate Delivery at Discount Prices



NORTH STAR Horizon® 2
32K Double Density
Assembled and Tested
List $3095
ONLY $2619
ASSEMBLED

HORIZON 1, DD **$2279**	32K, QD, List $2995 ..... **2539**
HORIZON 2, 32K, DD . **$2619**	48K, QD, List $4090 ..... **3469**
32K, QD, List $3595 ..... **3049**	64K, DD, List $3830 ..... **3239**
48K, DD, List $3590 ..... **3039**	64K, QD, List $4330 ..... **3669**

NORTH STAR APPLICATIONS SOFTWARE

(Exclusive for use with North Star Disk Systems — specify Double or Quad Density)

NORTHWORD, List $399	**$339**
MAILMANAGER, List $299	**249**
INFOMANAGER, List $499	**419**
GENERALLEDGER, List $999	**799**
ACCOUNTSRECEIVABLE, List $599	**499**
ACCOUNTSPAYABLE, List $599	**499**

NORTH STAR HARD DISK HD-18

18 megabytes, plugs into parallel port of North Star Horizon. Utilizes tried-and-proven 14" Century Data Marksman. List $4999. **OUR PRICE $4199**

NORTH STAR MDS-A — Double (or Quad) Density Disk System, Kit, List $799 . **OUR PRICE $669**
Assembled and Tested, List $899 **SPECIAL $719**

NORTH STAR MEMORY BOARDS

16K Dynamic RAM (RAM-16-A/A), Assembled, List $499	**$420**
Kit, List $449	**SPECIAL $299**
32K (RAM-32/A), Assembled, List $739	**$620**
Kit, List $669	**ONLY $499**

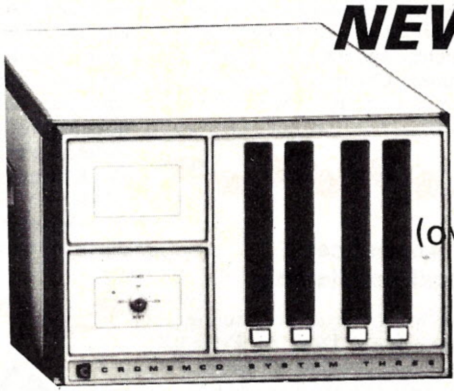
INTRODUCTORY SPECIALS ON ...

PREMIUM QUALITY BASF DISKS
CERTIFIED FOR QUAD SYSTEMS

(Box of ten)

5¼" DOUBLE DENSITY DOUBLE SIDED List$57.50	**$38.95**
8" DOUBLE DENSITY DOUBLE SIDED List$75.00	**$47.49**

Shipping $2.50 — Free Shipping In Multiple Of Two Box's

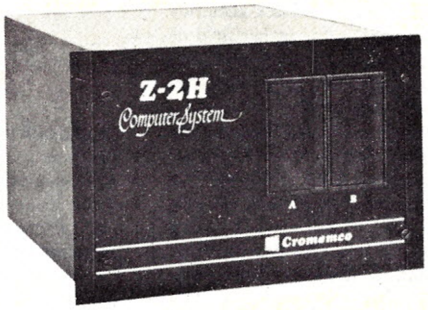


***NEW* System 3**
by CROMEMCO
Now with Dual Double Sided Double Density (over 2 megabytes of Storage) 64K of RAM
List $7395

LIMITED TIME INTRODUCTORY SPECIAL $6199!

CROMEMCO SYSTEM 2 — Now double Density with Double Sided Drives, Quad Capacity mini floppy disc drives. List $3990 .... **Only $3390**

CROMEMCO Z-2H



Full 11-megabyte Hard Disk system. Fast Z-80A 4 MHz processor, two floppy disk drives, 64K RAM memory, RS232 special interface, printer interface, and extensive software available.
List $9995
OUR PRICE $8489

NEW DOUBLE DENSITY CONTROLLER BOARD
From Cromemco
With built-in diagnostics — 16 FDC Controller
List $595 .... **OUR PRICE $505**

Z-2 COMPUTER SYSTEM List $995	**$845**
SINGLE CARD COMPUTER — SCC-W 4 MHz. List $450	**$382**
NEW COLOR GRAPHICS INTERFACE — SDI List $595	**OUR PRICE $505**

CROMEMCO HDD — 11/22-megabyte Hard Disk for use with existing systems. DMA controller. Transfer rate of 5.6 megabytes/second.

HDD-11, List $6995	**OUR PRICE ONLY $5939**
HDD-22, List $11,995	**$10,189**

SHIPPING AND INSURANCE: Add $15 or Horizons, $2.50 for Boards and Software. Hard Disk Systems and Cromemco systems shipped freight collect. Advertised prices are for prepaid orders. Credit card and C.O.D. 2% higher. Deposit may be required on C.O.D. All prices subject to change and offers subject to withdrawal without notice.

— WRITE FOR FREE CATALOG —

MiniMicroMart, Inc.

1618 James Street, Syracuse, NY 13203 (315) 422-4467 TWX 710-541-0431

master charge THE INTERBANK CARD
VISA

CIRCLE 157 ON READER SERVICE CARD

Northeast Computer Show

The Northeast Computer Show will be held at Hynes Auditorium/Prudential Center, Boston, MA, November 20-23, 1980. Show hours are: Thursday-Saturday, 11 a.m. - 9 p.m., Sunday, 11 a.m. - 6 p.m. Adult admission $5. An end-user public exposition featuring small and medium-sized business systems, scientific, engineering computers and small computers. Produced by National Computer Shows, 824 Boylston Street, Chestnut Hill, MA 02167. Tel: (617) 739-2000.

ICP Announces 3rd Annual Scholarship

To further the belief that a strong future begins today in the schools, Larry A. Welke, president of International Computer Programs, Inc. (ICP), has announced the third annual ICP Scholarship competition. The award will be made to a computer science or computer technology student for the 1981-82 collegiate school year. The scholarship will consist of one year's tuition plus education expenses up to a maximum of $5,000 paid to the U.S. college or university of the winner's choice.

The ICP Scholarship Committee is composed of highly qualified men and women from throughout the computer industry. They will base their selection on: the student's accumulative grade point average in his or her field of study; overall grade point average; need for financial aid; participation in data processing-related activities; school activities and leadership roles; and overall accomplishments and awards. An original essay will be the final test.

Applications will be available through the financial aid departments of most U.S. colleges and universities. The deadline for filing scholarship applications is November 15, 1980. For further information, contact Sheila Cunningham at ICP.

Mail applications to: Sheila Cunningham, Editor, International Computer Programs, Inc., 9000 Keystone Crossing, Indianapolis, IN 46240 (800) 428-6179.

Software Education Seminar

A two-day seminar, called "Designing Software Education" will be held in Boston, November 13-14 at the Sheraton Boston Hotel and in Washington, November 20-21 at the Sheraton Inn-Washington Northwest. The seminar will start promptly at 8:30 each morning and last until approximately 4:30 each afternoon. Developed for managers and training personnel charged with user and internal education, the seminar will focus on system documentation, classroom education, and cassette workbooks, plus video production (VTR) and computer-assisted instruction (CAI).

Describing low-cost, step-by-step procedures for developing education programs that

can be utilized early in the development process, the seminar will be taught by Mr. Shirley Mixon, author of AMACON's *Handbook of Data Processing Administration, Operations and Procedures,* and numerous articles on software education.

According to Mr. Mixon, "Software education is one of our industry's biggest problems. That's because few computer professionals know how to write well, develope training materials or instruct."

Registration is $495 per person. For more information contact Shirley Mixon Seminars, 4549-E Northside Parkway, Atlanta, GA 30339. Phone 404-955-3183.

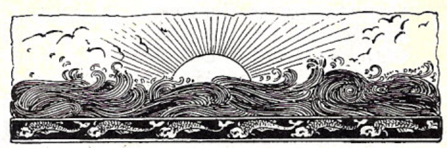
Call For Papers

The Mid-South Association for Educational Data Systems will hold their 8th Annual Conference February 26-28, 1981 at Richardson Towers, Memphis State University, Memphis, TN. Papers, workshops and demonstrations are being solicited in all categories of educational use of computers. Suggested topics are: Computer Aided Instruction (CAI), Computer Managed Instruction (CMI), Student Information Management Systems, teaching methodology, curriculum designs, user-producer communication, administrative applications and research developments.

Short round-table discussions on a variety of topics are also being planned for presentation. Please contact the Program Chairperson listed below if you wish to suggest a topic or serve as a round-table panel member.

Submit two double-spaced camera ready copies of an abstract or paper or two sets of documentation on workshops, demonstrations, or round-table presentations. All materials must be received by December 1, 1980, in order to be included in the conference proceedings.

Send contributions and program suggestions to: Lloyd D. Brooks, AEDS Program Chairperson, Office Administration, Memphis State University, Memphis, TN 38152. (901) 454-2453.



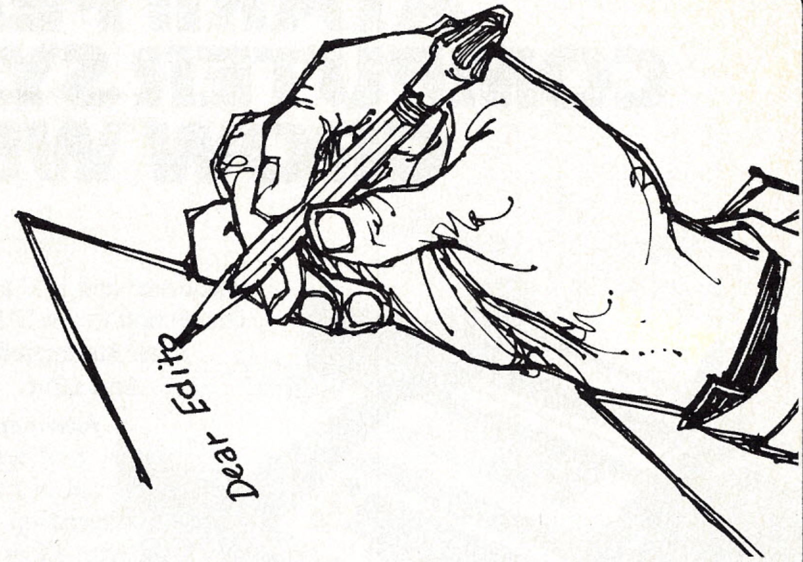
California Computer Swap Meet

The Fall, 1980 California Computer Swap Meet will be held on Saturday, November 8th, in Gateway Hall at Santa Clara County Fairgrounds (344 Tully Road, San Jose, CA) from 10 a.m. to 6 p.m. This semi-annual "happening" within the personal/micro computer industry is sponsored by John Craig, Publisher of *InfoWorld.*

This is where manufacturers, stores, and individuals come to unload their back rooms and garages with both top of the line and used merchandise. Whether you're looking for a diskette or a disk drive, an IC or a complete microcomputer system, application programs or games, books or magazines, or whatever . . . you'll find it there.

A special Consignment Table will be available for those who wish to drop off an item or two to be sold during the day. A free Literature Table is available to anyone within the industry. Admission to buyers will be through the purchase of a $5 Purchase Certificate, redeemable in full at any seller's booth, or by John Craig, for $3.50, if unused. Sellers, both individuals and companies, should call (415) 966-6546 (a friendly answering service) for booth prices, availability and reservations. Or, write to: California Computer Swap Meets, PO Box 52, Palo Alto, CA 94302. Having a good time at this event is not optional . . . it's mandatory.

Input/Output



Another Slice of Pi

Dear Editor:

In both TRS-80 Level II Basic and Applesoft, the inverse trigonometric intrinsic functions are limited to the arctangent. Both programming manuals offer "derived" functions for the arcsine and arccosine, using the arctangent function, as follows:

ARCSIN(X) = ATN(X/SQR(-X*X+1))
ARCCOS(X) = -ATN(X/SQR(-X*X+1))+1.5708

There are two problems which can arise when these formulae are used:

1. If $X = \pm 1$, a divide by zero error will occur, when in fact $X = \pm 1$ are legitimate values.
2. The principal value of ARCCOS(X) lies between 0 and π; the formula above (which is in fact based upon the ARCCOTANGENT) will give values from $-\pi/2$ to $\pi/2$.

The following formulae are suggested which allow all valid values of X ($-1 \leq X \leq 1$), and also return the principal values in the correct range:

(NOTE: PI = 3.141592654)

ARCCOS(X) = ATN(SQR(1-X*X)/(X+(1-SGN(X)*SGN(X)))) + PI/4*((1-SGN(X))*(1-SGN(X)))

ARCSIN(X) = PI/2-ARCCOS(X) (as shown above)

Each of these, written as single statements, can be used in a DEF FN statement, if desired.

The extensive use of the signum (SGN) function shows how powerful this function is for *selectively* including or eliminating terms from a formula.

Try it!

Michael P. Guerard
Associate Professor
Industrial Education and Technology
Glassboro State College
Glassboro, NJ 08028

Dungeons and Dragons

Dear Editor:

Bravo! Your Dungeons and Dragons article in your July edition was absolutely superb! I have been subscribing for about a half year and that issue is the pick of the crop.

I would also like to say that Glen Brannock's program made excellent use of the limited memory, but he didn't even mention some of the sub-classes of the ones listed. For all you computer-minded D and D buffs, by adding some extra "lineage," class number 1.1 could be a paladin, a type of holy fighter, 1.2 would be a ranger, a nature fighter, 2.1 could be an illusionist, a type of magic-user who deals with physical illusions, 3.1 could be a druid, a type of cleric who deals with nature, and finally 4.1 & 5 could be an assassin and a monk whose names tell their own professions. Also, a character race, the half-elf was omitted along with the gnome. All of these might be inserted by deleting some of the REM statements, although I couldn't do it with only 4K of RAM to work with on a Level II TRS-80.

Although these are minor details and Glen's program was superbly written, these character classes should have some mention.

In conclusion, I would like to say that I am 13 years old and have been playing D and D for quite a while and would like to contact other people regarding D and D and/or utility programs for the game.

Gary Katz
3 Noel Lane
East Brunswick, NJ 08816

P.S. My highest level character is a half-elf druid, thus causing me to write this letter!

Consumer Protection

Dear Editor:

I have been watching with some interest the controversy in some of your editorials, open letters, and the legal column, with respect to protection of software. I certainly agree that software authors deserve some protection for their work, else they would not bother to continue writing. (Unless they really had to do it. The names of the graphic artists and musicians who have been little compensated for their work but have continued to produce are legion.)

However, I have seen little from you or your magazine with respect to protection of software consumers. It is often difficult to get a demonstration of a particular piece of software prior to purchase. And if the product is other than a game (a word processing or income tax program, for example) several days are probably necessary to be able to evaluate the product to determine its ultimate utility in the desired application.

The "Smart" Video Terminal

ONLY
$675.00*

From the All-In-One Computer People:

You'll have to look a long way to find a better video terminal value. The H-19 Professional Video Terminal uses a standard typewriter-format keyboard, so you can start programming right away.

The 12-key numeric pad, in calculator format, makes entering of math data faster, easier. Eight user-function keys, 33 graphics keys give you extra flexibility. And 32 separate terminal functions can be controlled from keyboard or computer.

The 12" high-resolution CRT displays up to 25 80-character lines, features bright, clear readouts and reverse video capability.

Use the H-19 with any computer. All you need is an EIA Standard RS-232C Serial Interface — and you're ready to begin programming your computer system with this professional video terminal.

Get our FREE catalog — and find out why you should have the H-19 "Smart" Video Terminal.

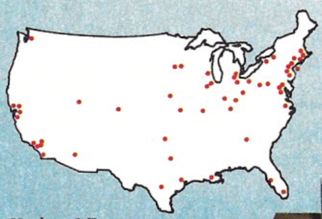
Heath®
Computer Systems
Dept. 355-714,
Benton Harbor, MI 49022

Send for
FREE CATALOG
describing the complete line of Heath computers, peripherals and software. Or pick up a copy at the nearest Heathkit Electronic Center.**

We won't leave you out in the cold

VISIT YOUR SERVICE CENTER

The "Smart" Video Terminal, along with the complete line of Heathkit computer products, is on display at the 62 Heathkit Electronic Centers throughout the U.S. and Canada. See your white pages for the nearest location.**



*In kit form, mail order, F.O.B. Benton Harbor, MI. Also available fully assembled and tested for only $995.00, F.O.B. Benton Harbor.

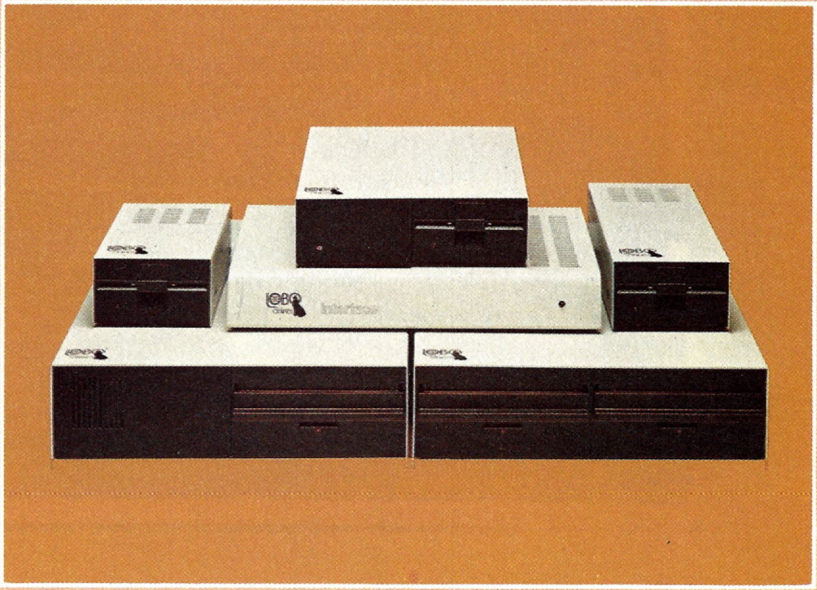
**In the U.S., Heathkit Electronic Centers are units of Veritechnology Electronics Corporation.

Prices and specifications subject to change without notice.

CIRCLE 146 ON READER SERVICE CARD

When It Comes To Add-on Memory...

LOBO Has It All.



LOBO DRIVES manufactures a full line of S-100 computer compatible disk drives. All drives are software compatible with most S-100 disk operating systems and applications software programs. Only LOBO DRIVES offers you the variety and choice of floppy and fixed disk drives. Choose from 5¼ and 8-inch floppies, 5¼ and 8-inch Winchester technology fixed disk drives, and several Floppy/Fixed disk combinations. Each LOBO DRIVES system is thoroughly tested and burned-in and has the famous LOBO DRIVES One Year, 100% Parts/Labor Warranty.

MODEL 400 5¼-INCH FLOPPY DISK MEMORY SYSTEM

A high-speed (298) Msec Access), high-reliability (8000 hrs MTBF), low-cost floppy disk memory system. It is available in both soft and hard sector formats, and a choice of single or double density configurations.

- Up to 220 KBytes Capacity
- Single/Double Density
- Soft Sector Format
- Complete Software Compatibility

MODEL 800/850 DUAL FLOPPY DISK DRIVE MEMORY SYSTEM

LOBO DRIVES offers you a choice of single-sided, single or double density (Model 800) or double-sided, single or double density (Model 850) dual 8-inch memory subsystems. Each system comes complete with chassis and power supply, cables, controller and interface.

- Compatible with Most S-100 DOS Systems
- Up to 3.2 MByte Capacity

MODEL 1850 DUAL FLOPPY/FIXED DISK MEMORY SYSTEM

No more worries about back-up. LOBO DRIVES has combined the latest state-of-the-art Winchester technology with the proven reliability and dependability of its Model 850 8-inch floppy disk drive to bring you the ultimate in memory expansion for your S-100 computer. The Model 1850 is the ideal memory system for small business and word processing applications.

- 5 or 10 MByte Fixed Disk Capacity
- 1.6 MByte Floppy Disk Capacity
- Software Compatibility
- Sealed Environment
- Winchester Reliability
- 70 Msec Average Access Time

MODEL 950 DUAL FLOPPY/FIXED DISK MEMORY SYSTEM

All the advantages of Winchester technology fixed disk memory: large capacity (6.38 MBytes), high speed (170 Msec avg. access time), and extended reliability, combined with the convenience of a built-in floppy disk back-up in one cabinet. Only LOBO can bring you the storage capacity of 16 mini-floppies at a fraction of the price.

- The Storage Capacity of 16 Mini-Floppies
- Built-in Back-up
- 170 Msec Access (Avg)
- Software Compatibility

See your nearest dealer, call, or write for the complete LOBO DRIVES story... find out just how competitively priced a quality drive can be.

LOBO drives INTERNATIONAL

935 Camino Del Sur
Goleta, California 93017
(805) 685-4546
Telex: 658 482

CIRCLE 204 ON READER SERVICE CARD

I/O, cont'd. . .

For example, this letter is being written using a well-known word processing program, which I must say does a reasonably good job for turning out a letter. The documentation (which I was not able to study until I got the package home) is decent, but the system has a number of quirks which must be worked around. I could continue on for several more pages, but the most interesting thing about the manual is the long attempt to disclaim warranties (which I did not see at all until I got the system home, and which certainly was not pointed out to me by the store which sold it.)

The manufacturer is not bashful about charging for the program. I can see why they do not want to be liable for consequential damages if the program reveals a hidden glitch only after thousands of dollars of input has been entered using it. (But what of the idiot who uses a $50 income tax program and gets socked with interest and penalties because of it?)

I believe that a consumer has a right to expect that the publisher and the retailer will stand behind the product, at least to the extent of returning the purchase price if the software does not perform to specifications.

The consumer should treat software publishers fairly. But fair treatment is a two-way street.

Barry D. Bayer
2842 Walnut Road
Homewood, IL 60430

P.S. One of the problems that I have noticed with my new word processing program is that I tend to get a bit wordy. Now is there someone who can do up a program to edit material that has been composed and formatted on the word processor . . .

Say "Cheese"

Dear Editor:

After typing in the Mouse simulation program by Jerry W. Lee in your July, 1980 issue, I was surprised to find my mouse went EEEK!, UGH, etc. but would not move after any length of prodding or waiting. After listing the program from the PDP-11 I used, I found that there were no errors at all in my program and all cursor commands worked. After playing computer, I came to the conclusion that the mouse could not advance past line 700 unless it sat, which brought it right back to where it started. But, after adding the following changes, I have enjoyed playing it, a friend getting the mouse to the goal in 26 seconds with prodding and 33 with none at all.

```
New line:
595 Q9%=0%
     ! RESET ILLEGAL BEHAVIOR FLAG
The major change:
700  H2%=H%+H1%
/    V2%=V%+V1%
/    N2=FNR2% IF H2%> =W4% OR V2%>=G% OR V2%<1%
/    Q9%=1% IF H2%>=W4% rOR V2%>=G% OR V2%<1%
/    N%=M%(N2%)
/    GOTO 590 IF Q9%=1%
```

Here, I set the flag Q9% to one if there is an illegal behavior, and conditionally branch back to 590 if the flag Q9%=1%, instead of going to 590 whether there is an illegal behavior or not. As you also can see, I reset Q9% in line 595 to avoid nasty errors.

As I said before, with these changes, MOUSE is a fun and interesting learning experience in conditioning.

Charles Congdon
Route 49 Swamp Road
Pittsfield, MA

I/O, cont'd. . .

Pavlov's Mouse?

Dear Editor:

I found an error in an introduction that you wrote for an article in *Creative Computing,* so now I have an excuse to write to you. I enjoy reading your magazine, and want to commend you for your interest in education.

The error is this: You said, in your introduction to "Building a Mouse" by Jerry W. Lee (July 1980), that "punishment, or negative reinforcement" does not work as well as reward (reinforcement) in changing an organism's behavior. You are quite right about the superiority of reinforcement, but the terms punishment and negative reinforcement are not synonymous.

Negative reinforcement is the *removal of an aversive stimulus.* Thus, a negative reinforcer serves to *increase* the probability of the response that it follows. For example, if a tight shoe hurts your foot, you are reinforced when you take it off. Negative reinforcement is the principal technique of the torturer: "Tell me what I want and I'll stop hurting you." In contrast, punishment is the administration of an aversive stimulus after a particular behavior. It serves to make that behavior occur less often. If you stick your finger into a parrot's cage and get it bitten, you will probably not put it there again.

Neil R. Carlson, Ph.D.
64 Harkness Road
Pelham, MA 01002

Thanks. We appreciate the, uh . . . punishment? — *Ed.*

Poli Sci I

Dear Editor:

Sorry to advise, but in the last issue of *Creative Computing,* the program "Basic Politics in America" has two serious bugs.

Old line data
3400 DATA 84,65,88,80,65,89,69,82
New line
3400 DATA 70,65,76,76,32,71,85,89
Old line data
6100 DATA 76,79,87,69,82,32,84,65,88,69,83
New line
6100 DATA 82,65,73,83,69,32,84,65,88,69,83

Hoping this "fix" will make the actions in Washington more clear to your other readers, and then your article and program will more accurately reflect our government's posture to those of us in the computer industry.

Paul Raymer
Paul's Electric Computer
POB 42831
Las Vegas, NV 89104

Effective Writing

Dear Editor:

Why do you have articles on effective writing in a computer magazine? I think they're great — love 'em — keep 'em comin — but I'm just curious.

Andrew L. Burt
7249 S. Vine St.
Littleton, CO 80122

Our answer is "Why not?" — *Ed.*

Vanity Unfair

Dear Editor:

I was happily typing in the music sight-reading program from *Creative Computing* June 1980, when I astounded to see the following line:

```
5465 V=V+194-PEEK(17315)-PEEK(17316)-PEEK(17317)-PEEK(17318)
```

In case you haven't looked at a TRS-80 memory map, these locations are right in the middle of Level 2 program text storage. This produces two problems:

1. The program must be typed *exactly* as it is written, or it will bomb.
2. The program will not run under Disk Basic.

Upon examination, I determined that these peeks referred to the characters "2000" used in a timing loop (line #45) that kept the opening title on the screen!

Normally, I find it easy to forgive (and change) a programmer's "cute" coding, but not in this case. When a published program contains a trap that occurs after almost 200 lines, it is a terrible diservice to the subscribers. This is doubly irritating when the trap seems to be there only for vanity's sake.

If anyone cares, the fix is to simply remove (or convert to a REM) line #5465. The program then behaves fine under Disk Basic, as well as Level 2.

Louis Shapiro
415 West 24 Street
New York, NY 10011

Adventure

Dear Editor:

Because of the recent popularity of Adventure games, I am interested in implementing one on the system that I use, an IBM 370. However, I am unable to use the published software I have seen since it is designed for home systems. Further, IBM Basic lacks the string handling functions supplied in home systems. Can you or any of your readers direct me to a version of Adventure written in standard Basic, Fortran, or IBM Call-OS PL/I? Thank you.

Timothy Marino
128 Partree Road
Cherry Hill, NJ 08003

There are versions of Adventure on some IBM timesharing systems, but tracking one down might be tough. Perhaps some of our readers can help you out. — *DL*

Errors Ill-Behaved

Dear Editor:

In my article, "A Consistently Well-Behaved Method of Interpolation," in the July issue of *Creative Computing,* attention should be called to some minor errors. In the last column on page 55, in the paragraph starting "The rationale," y_j should be y_j', and y_{j+1} should be y_{j+1}' (3 places). In the second paragraph of Conclusions, page 56, the first sentence should end with a right-hand parenthesis. In the 8th line of the last column on page 56, y_j should be y_j'.

Russell W. Stineman
Boeing Aerospace Company
P.O. Box 3999
Seattle, WA 98124

NEW PRODUCT

THE DATA FACTORY™

YOU CAN'T WORK HARDER, SO WORK SMARTER

This program is important to you. We at Micro Lab have tested them all. "THE DATA FACTORY" by William Passauer, is the most powerful data base system yet developed. It will provide instant accessibility to your records and files which you can then, rearrange in new combinations to give you information in seconds to make quick and accurate decisions.

OUR GUARANTEE

THIS PROGRAM WILL WORK. Micro Lab chooses to represent a very select group of professional programmers that meet our high standards for quality. Countless hours have been spent in our labs to insure these claims to you, and we back them with a contract to your dealer. Your program can also be updated if any new changes to improve this program are made.

MOST ADVANCED SYSTEM

The latest breakthroughs in a data base system have been incorporated into Bill Passauer's program. The unique new feature that sets it apart from all others is its complete modifiability. You may rearrange your data, removing part of it from the original disk, and form a new data base without reentering the data again. Add, delete, replace, or rearrange and compare fields or data at any time. Do an incredible 20 level search.

The program has no limitation to the amount of fields you can have within the limits of your computer. Your field lengths can have a maximum of 239 characters for each field and you can enter ANY CHARACTER as data (commas, quotation marks, etc.).

EASILY LEARNED

Any one can use it. The program prompts you as it runs. The easy to follow manual leads you through the set up of your data base and all the features. "The Data Factory" is organized in nine program modules. Only the module being used is loaded into memory to manipulate data, rather than the entire program. This saves memory for manipulating data rather than for program storage. There are so many other "common sense" features that set it apart from all others.

A UNIVERSAL SYSTEM

You may use "THE DATA FACTORY" at home or at work. Set up: Inventories, Mailing Lists (a printer is needed for mailing labels); Sales records; Accounts payable or receivable; Budgets; Library, recipe, or phone directories; Appointment calendar; Notices of subscriptions, license or warranty dates; Working or shopping lists, and many other applications that you will discover. All of the above can be accomplished from this one disk oriented program. No need to have separate costly programs for each purpose. With all the data on a disk, you can manipulate the information more easily and efficiently. Find any record using the record number, the data entered or any variation of the data in your data base. The from/to feature selects records by dates or amounts.

REQUIREMENTS

"The Data Factory" is presently being offered in APPLESOFT but will be available in other forms of basic shortly. Check with your dealer for other software varieties currently being handled. You will need 48k and Applesoft in ROM. "The Data Factory" is as powerful with one disk drive as with two. You do not lose any of its capabilities using only one disk drive. A printer is optional.

INSURANCE POLICY

Micro Lab is instituting a revolutionary new policy for the public: Micro Lab Disk Insurance.

You can feel secure. Two identical diskettes of the "The Data Factory" program will be included with the original purchaser's package. If you make a mistake and accidentally blow a disk, there is no time or money lost.

AVAILABLE NOW

"The Data Factory" is being offered nationally for the first time. It has been marketed and tested on a local level and has been received with a most enthusiastic response from both dealers and users.

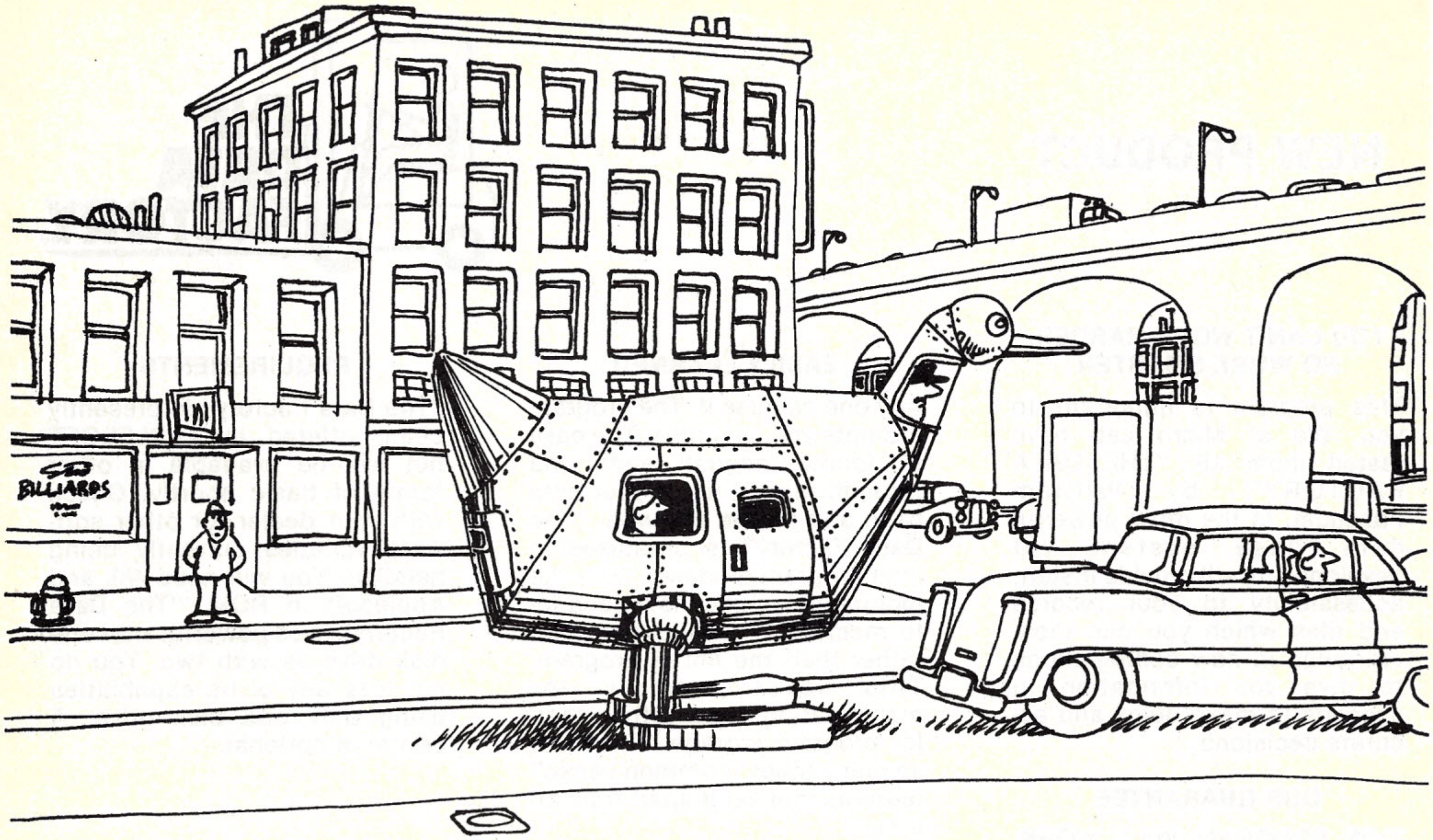
"The Data Factory" should be at your local Apple dealer now at an introductory price of $100.00. Stop in to see our demo disk.

micro lab

systems that work

811 STONEGATE • HIGHLAND PARK, IL 60035 • 312-433-7877

CIRCLE 156 ON READER SERVICE CARD



From "Tiny Footprints." Copyright B. Kliban 1978. Workman Publishing Co., New York. Reprinted by arrangement with the publisher.

The TRS-80™ Model III. A New Standard in Personal Computers!

The new standard is now here — this beautiful, feature-packed, one-piece desktop computer system at a very, very affordable price. Continuing the TRS-80 tradition begun with our famous Model I, the amazing Model III gives you everything you've always wanted in a personal computer—including easy expandability.

It Talks Your Language! Model III is available with either Level I or powerful new Model III BASIC. Best news of all is that nearly all Model I software is compatible with Model III, so you already have a huge library of applications to choose from. Radio Shack already offers over 80 quality packages — from games to sophisticated business programs to word processing.

Big Storage Capacity! Model III BASIC features dual-speed cassette loading (1500 and 500 baud). You can expand your Model III to support up to four (two integral) double-density disks at 175K each for a total system capacity of up to 670K bytes.

Radio Shack is Lowering the Cost of High Technology!

As Low As $699*

Powerful Memory! Up to 48K of internal memory is easily added, since no expansion interface is required. Model III is completely self-contained. Start with a 4K Level I system or move up to our 16K Model III BASIC right away for the applications you need.

High-Resolution Display! Every Model III has a sharp display of 16 lines of 64 characters. Model III BASIC adds lower case plus graphics and special characters.

Feature Packed! Every Model III includes a parallel printer interface and 65-key keyboard. Model III BASIC adds "extras" like a real time clock, scroll protect, keyboard controlled screen print, and RS-232 firmware.

And It's Very Affordable! The 4K Level I system is only $699.* The 16K version with powerful Model III BASIC is just $999.* So why wait, step up to the new standard now. Available at Radio Shack stores, dealers and Computer Centers everywhere.

Radio Shack®

The biggest name in little computers™

Send me your TRS-80 Catalog!

Radio Shack, Dept. 81-A-40
1300 One Tandy Center
Fort Worth, Texas 76102

Name
Street
City State
Zip Phone

CIRCLE 242 ON READER SERVICE CARD

*Retail prices may vary at individual stores and dealers. Special order may be required initially.

Random Ramblings

With David Ahl

The British Scene

While chips of the fish and chips variety seem to be the only ones successfully manufactored by the Brits, they seem to have done an amazingly nice job in taking our American chips and making them into computer systems.

Whilst on the subject of chips, I should mention that one mate stopped by our stand (booth) at the PCW show with a "floppy chip." This chip was about a foot long with 16 arms, stuffed with foam and covered with artificial fur and about the cutest rival you've ever seen for the attention of my family of animal puppets. (Yes, if you didn't know it I am a collector—of most everything, but most particularly of unique hand puppets, comic books (*Wonder Woman, Iron Man ROM, Machine Man, Micronauts* and others), T-shirts (which I wear, virtually exclusively), stamps, beer signs, and other stuff.)

By the bye (yes, it is "bye" and not by"), thinking about that last paragraph, it is an abomination. The *Wall Street Journal* recently had a piece about readability. They commented that the average letter and business report is virtualy unintelligible with a readability level of Grade 13 or higher. That sounds low (freshman college), but most people feel comfortable at a much lower level. I won't go through the discussion, but suffice to say that the average American prefers reading at the Grade 8 level. The average level of the WSJ is Grade 11, whereas a sampling of my writing is 11.7. Maybe a higher grade level seems intellectually appealing but this should not be held in esteem. The goal of this magazine is clarity to as many people as possible. What we strive for is interest, clarity and economy of words.

(Someday, when we have a bit of space to spare I will clarify our (my) philosophy and relate it to others like Freddie Laker, J.S. Bach, Ken Olsen and Big Bird et al. Stay tuned.

And now, on to the British scene.

A colorful aray of floppy belts in Petticoat Lane.

Tuscan S100

Tuscan: perhaps an unfortunate name for a computer. The TV comic bird and NJ milk company with similar names seem to give it an air of unreality. But it's quite real.

The Tuscan S-100 is a S-100 computer with excellent expansion capability.

The Tuscan S-100 is a unique cross between a single board computer and an S-100 unit. Effectively it is an S-100 mother board with Z-80 CPU, video, I/O and 8K RAM and EPROM built in with five S-100 slots for expansion. The case can house one or two 5-1/4" floppy disk drives and, of course, five S-100 cards.

Transam offers the Tuscan with a wide array of software including a resident 2K monitor or 8K Basic. A resident Pascal, disk Basic and CP/M are also available.

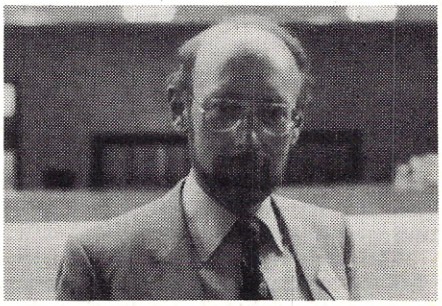
Prices for a Tuscan system are about the same as a similarly configured PET—about $1200 for starters in the U.K.—but seemingly the Tuscan offers much greater expansion capability.

Sinclair ZX80

Somehow, you hear the dimensions of this unit and say, "gee, that sounds small," but you don't really think about it. Well, do so—the unit is 6.85" x 8.58" x 1.5" and weighs 10.5 oz. That's small. Really *small* and *light*. It is slightly wider and an inch or two shorter than the standard tape cassette recorder you've got hooked to it. And it's considerably lighter.

Features: it's based on the Z-80 chip. Touch sensitive keyboard. Keyword inputs to Basic (like New, Load, Input, Print, List, Poke, Clear, Next, etc.). Graphics resolution of 48 x 64 pixels with 10 symbols plus space and inverse. Memory: 1K RAM built in, up to 16K add on promised (when?)

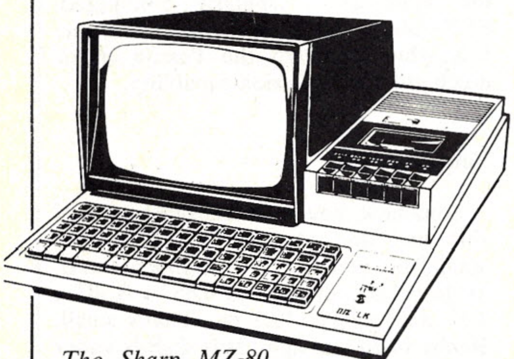
Coming up in the near future we'll have an in-depth review of the ZX80 and excerpts of a conversation with founder/president/designer Clive Sinclair. It looks like a winner to us but it's up to you to let us know what you think. U.S. price is only $200.



Clive Sinclair talked to us about his new ZX80 computer.

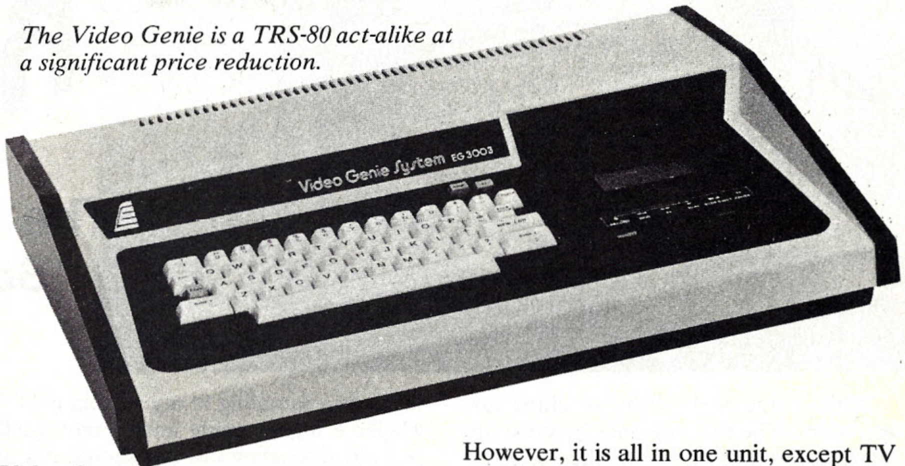
Sharp MZ-80K

Why this was not introduced at the summer CES we don't know, but it wasn't. In any event, it's neat. Z-80 based. Memory up to 48K. 50 x 80 pixel resolution (good but not outstanding graphics resolution). Fast cassette baud rate (1200 bps), and upper/lower case keyboard. Built-in real-time clock and 3-octave audio. Altogether a very well engineered and compact unit. Peripherals immediately available include floppy disk drives (128K each) and dot matrix printer (96 cps). It looks as though Sharp is serious. Very much so. System price with 48K, approx. $1000.



The Sharp MZ-80 is a well-designed, compact unit.

The Video Genie is a TRS-80 act-alike at a significant price reduction.



Video Genie

This is the TRS-80 act-alike that Personal Computer Corp, nee Microsette, showed at NCC in June 1980. Well, this, somehow, seems more real. Perhaps because we saw over 20 of them running. The only differences between the Video Genie and TRS-80 is that it (VG) does not have up and down arrow keys or a volume control on the cassette unit. However, it is all in one unit, except TV set/CRT/VDU.

In a derogatory way, we sometimes speak of those products as Chinese copies or Hong Kong copies. In this case, either is correct. The Video Genie is made by a Chinese company in Hong Kong and is an excellent functional copy.

Oh yes, one major difference is price. Video Genie costs about $700 for a Level II equivalent. P.S. Radio Shack dealers won't touch it. But can you blame them?

Acorn Atom

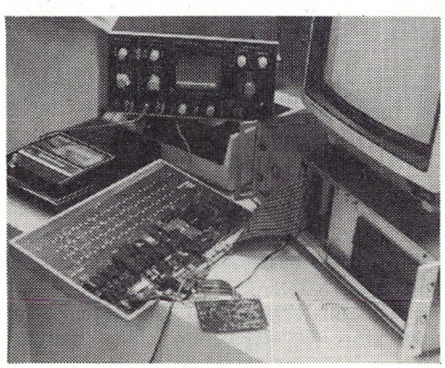
Some controversy surrounds the introduction of this machine. Certain people think it was introduced just a mite too soon after (or before actually) the Sinclair ZX80 (by a group of former Sinclair employees). In my humble opinion this seems unfounded since the Atom is quite different from the ZX80.

The Atom is based on the 6502 chip and has a full-stroke QWERTY standard sized keyboard. It measures 15" x 9-½" x 2-1/2". A 2K RAM and 8K ROM are included. The software includes an extended Basic with several powerful extensions such as variable length strings (up to 256 characters), the WAIT command, DO-UNTIL constructions, PLOT commands, DRAW and MOVE and several other extensions over standard Basic.

The Atom also has a nifty plug-in card with the 6809 MPU for those who want this unit instead of the 6502.

The basic Atom in kit-form costs about $300 or $400 assembled. More memory, peripherals and software are, of course, extra.

The Acorn Atom is a 6502-based computer with extended Basic and excellent graphics.



Even Atoms must be split occasionally.

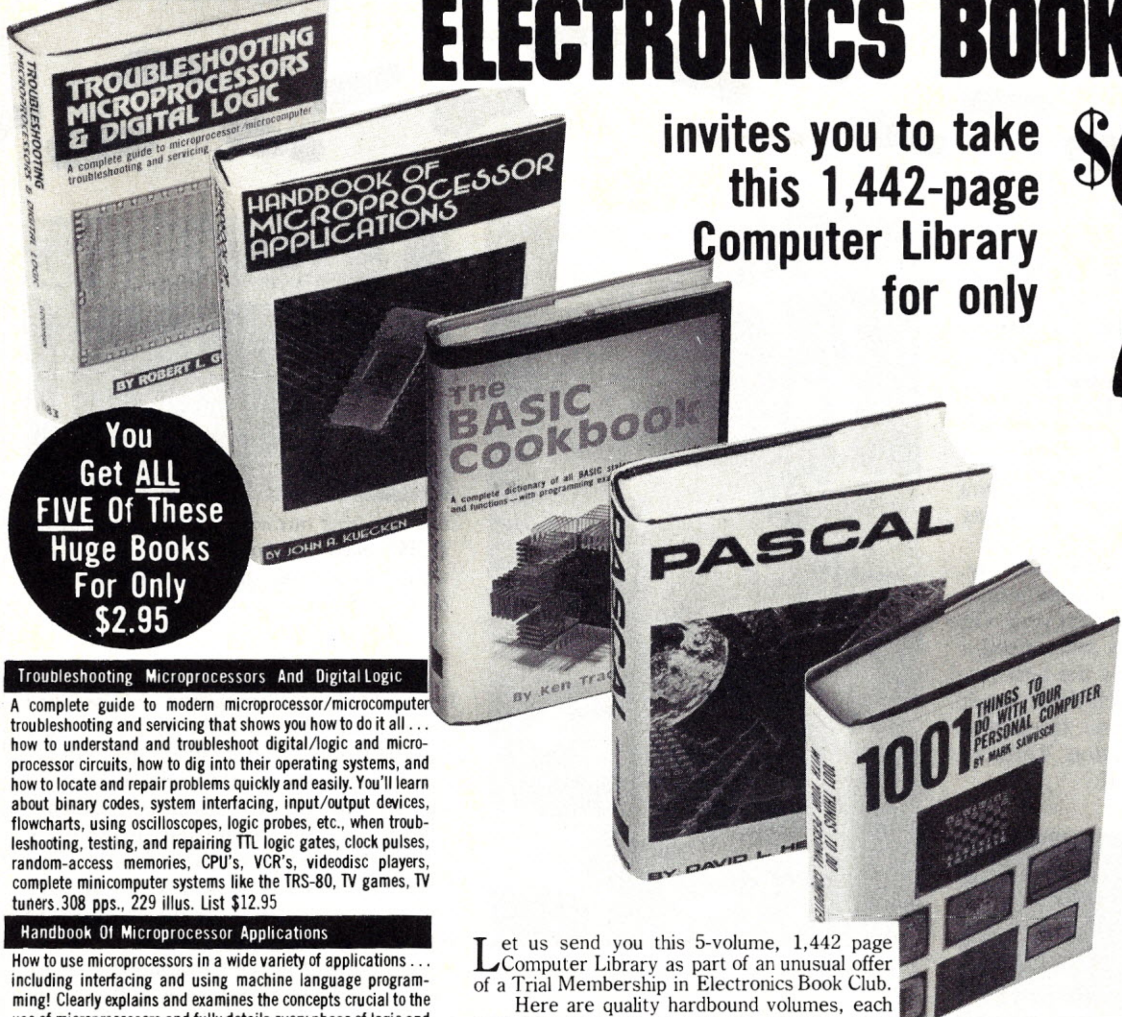
The PCW Show and World Computer Chess Championship

The 3rd Personal Computer World Show held in London from September 4 to 6 was well-attended all three days. Exhibitors were pleased with the quality of the attendance balanced between businessman and hobbyists. Many American companies were represented through agents, however *Creative Computing* was the only company with U.S. personnel. We were so pleased with the results that we've reserved a double stand (booth) in next year's show. For information about the 1981 show, write Personal Computer World, 14 Rathbone Place, London W1P 1DE, England.

An Extraordinary Offer to introduce you to the benefits of Membership in

ELECTRONICS BOOK CLUB

invites you to take this 1,442-page Computer Library for only $2.95



You Get ALL FIVE Of These Huge Books For Only $2.95

- ✓ Only $2.95 for ALL FIVE!
- ✓ Regular List Price $64.75
- ✓ Top-Quality Hardbinding
- ✓ Contains the very latest info on computers!
- ✓ Over 500 illustrations
- ✓ Contains over 275,000 words
- ✓ 1442 data-packed pages

Troubleshooting Microprocessors And Digital Logic

A complete guide to modern microprocessor/microcomputer troubleshooting and servicing that shows you how to do it all . . . how to understand and troubleshoot digital/logic and microprocessor circuits, how to dig into their operating systems, and how to locate and repair problems quickly and easily. You'll learn about binary codes, system interfacing, input/output devices, flowcharts, using oscilloscopes, logic probes, etc., when troubleshooting, testing, and repairing TTL logic gates, clock pulses, random-access memories, CPU's, VCR's, videodisc players, complete minicomputer systems like the TRS-80, TV games, TV tuners. 308 pps., 229 illus. List $12.95

Handbook Of Microprocessor Applications

How to use microprocessors in a wide variety of applications . . . including interfacing and using machine language programming! Clearly explains and examines the concepts crucial to the use of microprocessors and fully details every phase of logic and machine decisions: Boolean algebra, the truth table, OR, NOR, AND and NAND functions, etc. Learn to document and analyze a problem, locate any given step, calculate forward jumps, use timing loops, calculate delays and more. State machines, input/output functions, data buses, ROM and RAM. This book is practical . . . and focuses on the how-to of using microprocessor chips. 308 pps., 94 illus. List $14.95

The BASIC Cookbook

A complete dictionary of all BASIC statements, commands, and functions—with programming examples and flowcharts—thoroughly defines the BASIC vocabulary, illustrates the definitions with sample programs, and clarifies the programs with matching flowcharts. You'll learn how to use each BASIC term in a workable program. ARRAY, COS, END, FOR-TO, GOSUB, INPUT, LIST, RANDOMIZE, REM, SCRATCH, SGN, SQR, TAN. It also defines programming terms that apply to APL, ALGOL, COBOL, FORTRAN, RPG, PL1, and other languages. 140 pps., 49 illus. List $7.95

PASCAL

A programmer's guide to using Pascal, Tiny Pascal and Supersoft Tiny Pascal . . . including actual programs and helpful exercises! Starting with how to load a Tiny Pascal cassette into a TRS-80 system, goes through all the steps necessary to become proficient in this new language. Learn to read syntax diagrams; use WRITE statements to print characters and do TRS-80 graphics; enter integers with READ statements; use logic with AND, OR and NOT, etc. You'll also find out how to put together complete READ and WRITE programs, and use repetitive (looping) statements to write unending loops . . . plus how to "goof-proof" entries. 350 pps., 106 illus. List $15.95

1001 Things To Do With Your Personal Computer

Over 1,000 time-saving, money-saving, effort-saving and just-plain-fun applications—with actual programs, printouts, flowcharts, diagrams and illustrations. Twelve Chapters contain programs for any use and taste, and applications for everyone: business and financial mathematical, technical and scientific, educational, statistical, control and peripheral, hobbies and games. Includes a shorthand translator, weather forecasting, precise values for camera settings, animated films, model railroads, controlling household devices like wood stoves, Morse code, almost 100 games. 336 pps., 100 illus. List $12.95

Let us send you this 5-volume, 1,442 page Computer Library as part of an unusual offer of a Trial Membership in Electronics Book Club.

Here are quality hardbound volumes, each especially designed to help you increase your know-how, earning power, and enjoyment of electronics and computers. Whatever your interest in electronics/computers, you'll find Electronics Book Club offers practical, quality books that you can put to immediate use and benefit.

This extraordinary offer is intended to prove to you, through your own experience, that these very real advantages can be yours . . . that it *is* possible to keep up with the literature published in your areas of interest, and to save substantially while so doing. As part of your Trial Membership, you need purchase as few as four books during the coming 12 months. You would probably buy at least this many anyway, without the substantial savings offered through Club Membership.

To start your Membership on these attractive terms, simply fill out and mail the coupon today. You will receive the 5-volume Computer Library for 10-day inspection. YOU NEED SEND NO MONEY. If you're not delighted, return the books within 10 days and your Trial Membership will be cancelled without cost or obligation.

ELECTRONICS BOOK CLUB, Blue Ridge Summit, Pa. 17214

Facts About Club Membership

- The 5 introductory books carry a publisher's retail price of $64.75. They are yours for only $2.95 for all 5 (plus postage/handling) with your Trial Membership.
- You will receive the Club **News**, describing the current Selection, Alternates, and other books, every 4 weeks (13× a year).
- If you want the Selection, do nothing; it will be sent to you automatically. If you do not wish to receive the Selection, or if you want to order one of the many Alternates offered, you simply give instructions on the reply form (and in the envelope) provided, and return it to us by the date specified. This date allows you at least 10 days in which to return the form. If, because of late mail delivery, you do not have 10 days to make a decision and so receive an unwanted Selection, you may return it at Club expense.
- To complete your Trial Membership, you need buy only four additional monthly Selections or Alternates during the next 12 months. You may cancel your Membership any time after you purchase these four books.
- All books—including the Introductory Offer—are fully returnable after 10 days if you're not completely satisfied.
- All books are offered at low Member prices, plus a small postage and handling charge.
- **Continuing Bonus:** If you continue after this Trial Membership, you will earn a Dividend Certificate for every book you purchase. Three Certificates plus payment of the nominal sum of $1.99 will entitle you to a valuable Book Dividend of your choice which you may choose from a list provided Members.

ELECTRONICS BOOK CLUB

Blue Ridge Summit, Pa. 17214

Please open my Trial Membership in ELECTRONICS BOOK CLUB and send my 5-volume Computer Library, invoicing me for only $2.95 plus shipping. If not delighted, I may return the books within 10 days and owe nothing, and have my Trial Membership cancelled. I agree to purchase at least four additonal books during the next 12 months after which I may cancel my membership at any time.

Name ____________________ Phone __________

Address ______________________________

City ______________________________

State ____________________ Zip __________

(Valid for new Members only. Foreign and Canada add 15%.) CRC-1180

CIRCLE 196 ON READER SERVICE CARD

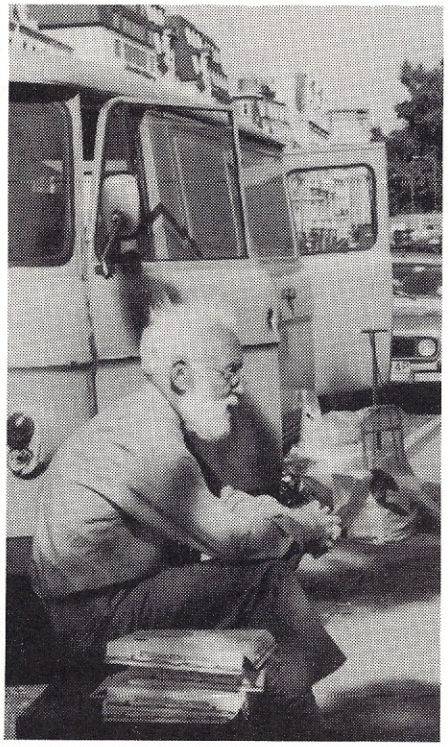
One of the exciting events of the show was the World Microcomputer Chess Championship. Held adjacent to the hotel bar, the tournament had an air of conviviality not normally associated with chess tournaments. David Levy, the tournament master, promised us a complete review of the tournament in his column. Not to steal his thunder here, but you ought to know that Chess Challenger was the winner.

However, if David Levy has his way, there will be a new chess champion (machine variety) on the scene next year. On September 10 his company announced a chess machine—supposedly the best yet. For more information watch these pages or, if you can't wait, write Intelligent Software, Ltd., 104 Hamilton Terrace, London NW8 9UP, England.

Is there any doubt that this fellow is one step ahead of us all?

Graphics and Animation

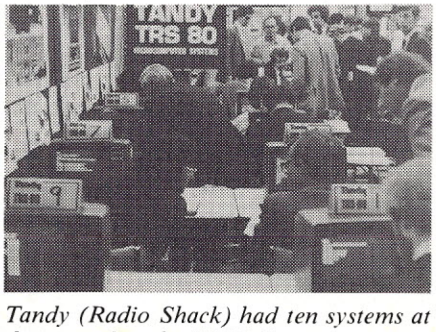
Following a delightful Italian dinner in a small restaurant just off Russell Square, we adjourned to John Lansdown's flat for a viewing of some of his latest computer animations.



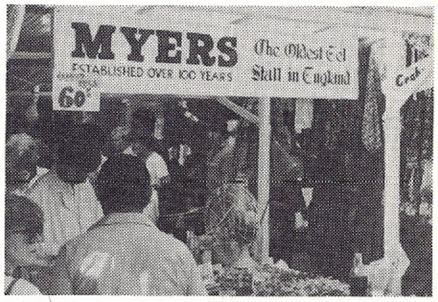
Contemplating one's stand on Picadilly by Green Park.

These animations were done for Spanish National Television and dealt with energy conservation (a concern not just in the U.S.). Unless one knew that these animations were done by computer, the difference between the computer and conventional drawings is practically undetectable.

Indeed, Alan Sutcliffe and John told us that when they did the landing sequence in *Alien* the director objected that it was "too realistic" and didn't look as though it were done by a computer. In the final version, the computer animation was degraded by displaying it on a lower resolution terminal and then further degraded in the filming so that the audience would believe it was done by computer.



Tandy (Radio Shack) had ten systems at their stand at the PCW show.



No, this was not one of the stands at the PCW show, although we're sure they would have done quite well if they were there.

The irony of the situation is that in the future computer animation will be even better than it is today. By catering to the current beliefs of audiences, the film actually portrayed a less realistic view of the future than could have been shown.

Alan Sutcliffe, by the way, is the founder of the Computer Arts Society and John Lansdown is secretary. Both have promised us material for future issues to which we greatly look forward.

Membership in the Computer Arts Society costs $5 per year for which you get a few issues of *PAGE*, the society's bulletin. For U.S. membership, send $5 to Kurt Lauckner, Math Dept., Eastern Michigan University, Ypsilanti, MI 48197.

Everyone has a chance to sound off at Speaker's Corner, Hyde Park.

Best massage your data ever had.

Data Wrangler is a comprehensive file handling program for the Apple II* computer. Its unique design puts a whole new level of information management skills at your fingertips, whether you're a novice user or an experienced professional.

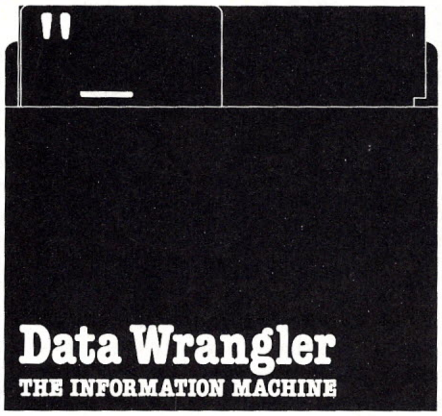
Information enters the system from any input device recognized by your hardware configuration—keyboard, disk, cassette, remote computer via telecommunications, you name it.

Once in, the data occupies a workspace as large as all of free memory in your system. It is in this workspace, at machine-language speed, that your information can be listed, edited, transformed, replaced, manipulated, searched, sorted, formatted, or otherwise changed to your precise requirements.

Most major functions are accomplished with a single key. At any time, all the information in a file can be printed to an external printer, saved to the disk storage device, sent to another computer via telecommunications.

You are in total control at every step. The program is command driven, rather than menu driven, giving you much more speed and flexibility in real time operation once you master the terms. A hefty documentation package takes you step by step, clearly and simply, through the entire resources and capabilities of the package.

What's more, the program—while copyrighted—is completely copyable. It is anticipated that an active and imaginative user group will



emerge, in which the members will share new applications and program notes with each other, for increased satisfaction and value.

What you can do, for instance...

1. Activate your system with the ANSW command before you leave home. Call it up from a remote location and operate it as if you were there.
2. Call a remote time-sharing system such as the Dow Jones News and Quote Service. Get the information you want and save it to disk. Edit, sort, search, and get the particular nuggets you want, off line.
3. Choose a character of your choice as a file separator. Have it indicate a carriage return, a tab move, a column formatting device. Use it as a sort key.
4. Make new files that other programs can use, in binary or text.
5. Create a file to include both visible and suppressed data. Have the printer print part of it, keep the rest of it in storage but not printed out.
6. Use Data Wrangler as a text editor, with full search and replace features.
7. Write a program in any language and save it as a binary or text file. Upload it to a timesharing service and debug it on remote. Download the corrected program and save it for use later.
8. Read in a binary or text file from disk, perform editing functions, save it back to disk, revise as you like.
9. Save graphics or any other image from the hires screen as a binary file. Use the graphics table for input. Make a sequence of images part of a CAI program, as a slide show.
10. Simulate program operation for demos by putting up menus, followed by hires charts or graphs.
11. You name it.

TO ORDER "Data Wrangler," THE INFORMATION MACHINE, send your check or money order in the amount of $75 to The Computer Room, 106 E. Oak St., Chicago, IL 60611.**

*Trademark of Apple Computer, Inc.
**You are free to make as many copies as you like for your own use. Seller, however, cannot accept return of the program disk or manual for refund, for any reason whatsoever.

THE COMPUTER ROOM

An Experience Most English

Betsy Staples

The first official British excursion by Creative Computing staff members had a totally uncomputerized beginning. The employee at the Air India ticket counter at Kennedy airport had what he insisted was an indeterminate number of standby seats for sale. How he knew whether he had any at all was a mystery since the only concessions to modern technology in evidence on his desk were a telephone and pocket calculator. Writing tickets by hand was a painfully slow process — it took nearly two hours to process the 12 people ahead of us.

"Who's next?" he said while writing the ticket of the man in line ahead of us. "I have one more ticket." Pretending not to hear, we gave him two passports and he wrote two tickets. At times like that one does not ask questions.

Upon arrival in London after a late, but otherwise uneventful flight, we were immediately appalled by the high price of just about everything. The bed and breakfast hotel near Earl's Court where we stayed was a far cry from the bargain lodging we had come to expect on previous trips.

After a short nap to keep jet lag at bay, we set about the task of locating the cartons of books and magazines we had shipped to England for the Personal Computer World Show. Due to human error only 36 of the 65 cartons we sent by sea in early July had arrived, and the remainder had been sent by air at the last minute. No one in H.M. Customs had noticed that there were only 36 cartons, and clearance had been granted for all 65.

The boxes that came by sea were cleared by humans at Felixstowe, and the ones that arrived by air were subjected to computer scrutiny in London. Naturally, the two processes were incompatible, but after many phone calls and much gnashing of teeth the books were delivered to the Cunard Hotel where the show was to be held.

The same H.M. customs organization on the same day opened another box and found the 50 wind-up robots we planned to offer as premiums to people who entered subscriptions at the show. They insisted that the box contained "electronic games," and it took us some time to convince them otherwise.

After we were sure that all was in order for the show, we set out to do some sightseeing. We visited the British Museum of Science and Technology where we saw some very impressive machinery — giant steam engines and the like — and an exhibit devoted to the history of computing. They even had a display that allowed us to see bubble memory in action.

The next day our peregrinations took us to The Game Centre where Graham Levin maintains an incredible stock of games — everything from dice to computers. We're sure certain Dungeons and Dragons fans of our acquaintance would have gone bananas over the selection of books and markers for that game alone.

Back at the Cunard for set-up we found very friendly, helpful show staff and union workers. The British union people apparently have enough work that they don't feel compelled to spend their time thwarting the move-in and set-up activities of exhibitors. They were, however, insistent upon their tea: at precisely 4:00 p.m. one of them arrived with a tray of cups (not paper) and a pot of tea. They all left what they were doing and sat on the floor drinking and chatting.

When the show opened, our agent in the U.K., Hazel Gordon, was there to help at the stand. The first thing she told us was that we were not to accept personal cheques unless the customer could produce a "banker's card." We soon learned that British banks issue to deserving depositors a plastic card which not only identifies the bearer but guarantees his or her check up to £50 — a splended system that U.S. banks would do well to imitate.

The show was a great success. We found the British computer buffs charming and eager to learn what was going on across the Atlantic.

Sunday, the day after the show, we spent in totally un-computer-related activity roaming the streets of London. Petticoat Lane, a giant flea market offering everything from clothing to toys to jellied eels; a craft display in Greek Park and a walk through Hyde Park, including Speakers' Corner, were all on our itinerary. The weather was quite warm and sunny, and Londoners thronged the parks to take advantage of it.

Tuesday morning was devoted to retrieving and sending several cartons of leftovers from the show to Hazel in Nuneaton, a very small town in the Midlands. The personnel in the British Rail terminal provided the same efficient, good natured service we had come to expect by that time, and soon our books and catalogues were on their way, along with a hot air balloon (deflated) and a small boy's trunk for boarding school.

Tuesday afternoon we elected a visit to the London Zoo, "thought by many," according to our guidebook, "to be one of the finest in the world." We found the admission price (£2.50 — about $6) a bit steep, but the zoo itself was delightful. The animals are housed in a curious mixture of antiquated "houses" and cages and up-to-date areas where a moat is the main deterrent to the mingling of guests and residents.

One of our favorite exhibits in the zoo was the Insect House. Many varieties of unusual insects (some of them venomous) are kept in terrarium-like environments with glass fronts which allow visitors to get a good look at them and their activities. Have you ever watched a black widow spider wrap up next week's dinner?

And, of course, we saw the giant pandas, who were a bit lethargic — perhaps because of the heat. The charm of the zoo is heightened by the well-kept gardens located throughout and the canal which runs through it.

After the zoo, we took the underground to Victoria Station where, after standing in a queue for about 15 minutes, we purchased tickets for Freddie Laker's Skytrain back to New York. A computerized system, pleasant personnel and a DC-10 that stayed aloft brought to a conclusion our most delightful English Experience. □

electronic toys and games

David Lubar

Once, there was Pong. Then home arcades flooded the market. Soon, consumers had a large selection of electronic toys and games from which to choose. This year is even better. The products introduced at the Toy Fair and Consumer Electronics Show run the gamut from new, exciting toys to rehashes of last year's big sellers. This month and next, we'll take a look at some of these toys. A few non-electronic products will be included on the basis of their being innovative, clever, or just plain fun.

Tri-1 from Fonas

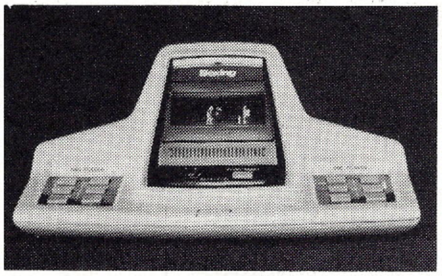
All-Star Baseball, Batting Champs, and Star Chase form the trilogy populating this hand-held game. First, a compliment on the buttons. They are sturdy and press with a nice click. The games are done with moving dots of light. Baseball has all the expected features, with two players alternating turns at bat. There is also a button that allows the player to attempt to steal. The game can handle extra innings if there is a tie at the end of regulation play. One player can, of course, play both sides. The sound produced by the unit is loud, but not annoying. Star Chase gives the player two ships which he must land on two enemy ships. Each ship can be moved horizontally or vertically, but the direction of the up-down and left-right moves is random. The game is a bit difficult to figure out from the instructions, but fun once you understand what is happening. Another nice touch is the inclusion of instructions on the back of the unit.

The Generals from Ideal

The Generals is the national game of the Philippines. It was introduced to this country a while back in the form of Stratego, and became quite popular. Ideal's version adds a new touch. In the game, each player has a number of pieces of different rank; generals with one to five stars, privates, agents, a flag, and others. The object is to capture your opponent's flag or get your flag to his back rank. When one piece attacks another by moving onto an occupied square, the higher ranking piece wins. In the old version, you could use low-valued pieces to attack, thus revealing your opponent's piece. In Ideal's game, the players place the two pieces on a special section of the board, then flashing lights indicate the winner. This adds a new level of strategy to the game. Those who liked Stratego should consider the Generals. The game can be quite fascinating.

Boxing, Space Laser Fight, and Classic Tones from Bambino

Bambino has one of the best new lines of the season. Their graphics are outstanding, and the games are well designed, with high-quality plastic and contoured buttons. Boxing is for one or two players.



The control of the boxer includes stooping and standing up, punching at a high, middle, or low level, and backing up. A boxer who is knocked down might be able to get back up, but the more hits he receives, the better the chance he'll be down for the count. Playing against the computer results in a challenging game, but the player stands a good chance of winning at level 1. Space Laser Fight is another game of combat for one or two players. Here the players jump, stand or stoop while firing at each other. The lasers can be fired at three levels, and points are scored for hitting your opponents legs, arms, or head. A head shot also results in temporary electronic disintegration. Random obstacles move between the players, adding to the difficulty of scoring

a shot. At level 1, the computer is a tough opponent. At level 2, it is very difficult to win. Again, the graphics design is excellent and the game enjoyable.

Bambino also sells a series of small electronic organs, called Classic Tones. The one we received had several nice touches, including a jack to allow for voice input to the speaker through a microphone in case the kiddies want to sing along with their playing. There are two knobs on the organ, one for volume and one which controls a metronome, allowing for a steady beat behind the music. A songbook and color-coded keys get the beginner started right away. Unfortunately, this is a single-voice instrument; chords can't be played. Still, it can serve as a good introduction to music for a child.

Delux Football and Hippo Lot-O-Fun from Tiger

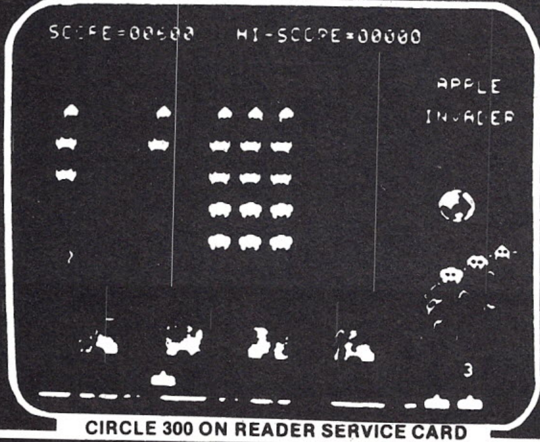
Football comes with an interesting control button. It's a four-way contoured control that is used for selection of plays and movement of players. One or two players can play, and there are several skill levels. The higher levels use more men and move at a faster pace. After the player selects a running or passing play, he moves one of the men down the field, either running with the ball or running for a pass.

SPACE WAR

You're in command in **SPACE WAR!** Destroy your opponent's ship by forcing him to collide with the sun or to explode upon re-entry from hyperspace ... or challenge him face to face with missile fire. You're in command of the speed and direction of your ship. You control the timing of your missiles. You select the game mode from five options, including Reverse Gravity, and the battle begins. Accelerate to place your shots--and escape into hyperspace before your opponent comes within range. But be wary, he (or she!) may circle out of sight and reappear on the opposite side of the galaxy! (This is the classic MIT game redesigned especially for the Apple.)

and SUPER INVASION

- **Super Invasion** is the original invasion game, with the original moon creatures and faster action than any other invasion game.
- Features superb high resolution graphics, nail-biting tension and hilarious antics by the moon creatures!
- Self-running "attract mode" of operation for easy learning and demonstrating of the game.
- As good in every way as the famous Invaders arcade game.
- High speed action ! • Sound effects!
- Runs on the Apple II and the Apple II Plus



Fifty-five aliens advance and shower you with lethal writhing electric worms. As you pick off the aliens, one-by-one, they quicken their descent. They whiz across the screen wearing away your parapets, your only defense, coming closer and closer to your level. **Super Invasion** is the **original** invasion game with the original moon creatures and faster action than any other invasion game on the market.

Super Invasion is available for only $19.95 on cassette (CS-4006) for a 32K Apple II. **Space War** is $14.95 on cassette (CS-4009) for a 16K Apple II. **Space War** and **Super Invasion** are on one disk (CS-4508) for a 48K Apple II for only $29.95.

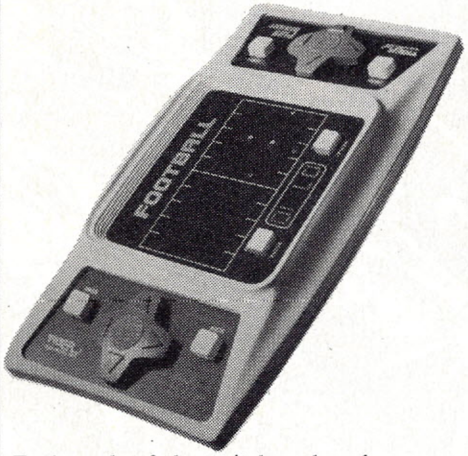
Send payment plus $1.00 shipping and handling to Creative Computing Software, P.O. Box 789-M, Morristown, NJ 07960. NJ residents add $1.00 sales tax. Bankcard orders may be called in toll free to 800/631-8112. In NJ call 201/540-0445.

sensational software

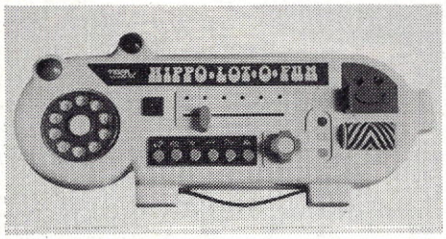
creative computing software

CIRCLE 300 ON READER SERVICE CARD

Toys & Games, cont'd . . .



Each push of the switch makes the man move one space. This constant pushing can become a bit tiresome. The game also provides a half-time show with marching dots forming various patterns.



Hippo Lot-O-Fun should definitely amuse the youngsters. A space-age crib toy, it has lots of buttons and knobs, all of which produce music. The sound might be a bit too quiet, but most parents can be glad about that. A nice touch is the interrupt feature. Any time a button is pressed, the hippo goes into action; the child doesn't have to wait for the previous activity to finish.

Le Boom from Lakeside

Lakeside's Le Boom is a great game for one or two players. The game is a bomb. Really. A large plastic bomb that will blow up if you don't play the game correctly. There are four games inside Le Boom, with several options. Instead of

buttons, there is a sheet of heavy plastic with circles on it. The buttons are beneath this. In all of the games, clues to the correct sequence to push are given with sound. As time grows shorter, the bomb's hum rises to a whine. If you make a mistake or run out of time, Le Boom emits the sound of an explosion. But if you succeed, the bomb winds down and plays a victory tune. In one game, you have to press the circles until one of them emits a high-pitched sound. That circle can't be touched again, under penalty of blowing up. This continues until six such circles have been discovered. The other games involve discovering a secret circle, avoiding previously played circles, and finding the right combination of three key circles. All in all, a nice game.

Flash from Ideal

This is an electronic bean-bag target game. The unit can be placed on the floor or hung from a wall. In the center is an LED which displays points and other information. Several players can participate, and there are five games to choose from. The unit is a bit loud, so lock up the batteries after dinner. The games are fairly clever. One involves hitting as many as possible of the segments as they light up in turn. Another displays the point value of each segment for an instant. Then the players have to remember which are the high-scoring segments. Flash would be good at parties since the games are loud and don't last too long.

Electronic Computer Bowling from Vanity Fair

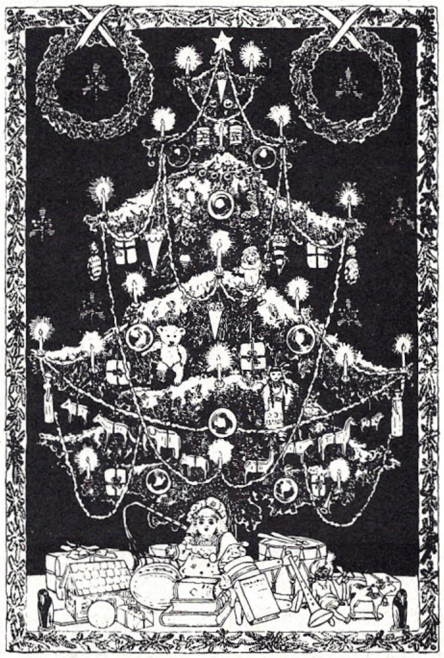
The game is an enclosed alley with lights for pins and ball. The player releases the ball by pressing a bar. Releasing the bar will cause the ball to hook. The game, which is for one or two players, is interesting, though it doesn't take long to master it. Still, children might enjoy it, especially when played in competition.

Super-Sonic Electronic Mastermind from Invicta

This compact unit plays a numerical version of this well-known game with from three to six code numbers. The unit comes in an attractive plastic binder that holds the game and a pad that is used to keep track of guesses. For anyone who has lived on a deserted island for the last few years, Mastermind is a game of code breaking.

The player makes guesses as to the numbers and their positions in a secret number. He receives clues informing him of how many numbers are correct but in the wrong position, and how many are in the right position. The electronic version keeps track of the number of guesses and the time spent making those guesses. There is an option that allows one player to enter a number which the other must guess.

Next month, more toys, including a hand-held version of Invasion, more great graphics from Bambino, a drag race simulation, and lots of other goodies. □



Man-to-Man Combat for the Apple

Randy Heuer

Computer Ambush is the second in a series of computer conflict simulations from Strategic Simulations Inc. (SSI). The first simulation of the series, *Computer Bismarck* was reviewed in *Creative Computing* (July, 1980). Like its predecessor, *Computer Ambush* is a simulation of combat in World War II. However in this game, the scale of combat is brought down from that of fleets and aircraft squadrons to the individual infantry soldier in France, 1944. Human players take the role of a squad leader. The time scale of the simulation is of tenth's of seconds with each square on the map representing three meters.

These individual games range from ambushes and raids to design-your-own formats.

All of the elements of man-to-man combat are brought into play, so some people may find this level of conflict simulation too graphic to stomach. Knives, bayonettes, automatic weapons, grenades, plastic explosives, leadership, panic and tactical planning are all part of the game. Decisions each player must make as squad sergeant or German "feldwebel" must account for these features. In addition, each soldier in a player's squad is an individual. Each is numerically rated for weight, strength, intelligence, dexterity, powers of observation and marksmanship. Thus PFC Charles Lawson may have exceptional powers of observation and dexterity, but rather poor ability to handle his weapon. Similarly, Klaus Muller may possess great strength, but lack common sense. Therefore it is up to the human player to take advantage of each soldier's attributes in order to eliminate the enemy troops or survive an ambush.

Randy Heuer, 5 Dogwood Road, Morristown, NJ 07960.

On to the Program

Computer Ambush is designed to run on a 48K Apple with Applesoft in ROM and at least one disk drive. Included in the game box are two sets of plastic coated maps, two information cards, a pair of grease pencils, a 14 page instruction booklet and the diskette. At the start of each game, the player (or players) select one of three solitare (one human player vs. the computer) or four two-player scenarios. These individual games range from ambushes and raids to design-your-own formats. Each player may then select a number of soldiers from his ten-man squad to use in this game. Generally speaking, it's wise to limit the number to four or less on each side, as many soldiers require a great deal of computation time. More on this problem later.

Each player then supplies orders for each member of his squad to execute during the next turn (approximately 3-5 seconds of simulated time). A Hi-Res map of the field of play is displayed along with the positions of the individual members of the squad and any "spotted" enemy soldiers. The player then gives each of his soldiers a list of commands to be executed in the turn. Typical commands are walk, crawl, run in a given direction, fall down, stand up, prepare weapon, fire weapon at an individual square or into an area if an enemy soldier appears, prepare grenade, throw grenade, etc. Each of these orders require a finite amount of time and energy to complete, and after an individual soldier has used up all of his time units for that turn, he can do no more.

After a player has given a set of orders to all of the members of his squad, the other player (or the computer) does the same for his troops. When both sides have completed providing orders for this turn, the computer then executes both sides' orders simultaneously. This step unfortunately requires a substantial amount of time, as much as a half an hour or more when simulating a five second turn. Although this calculation time can be reduced somewhat by playing with an option that allows spotting of the enemy at all times, it still means that players spend a significant portion of their time waiting for the computer.

It is up to the human player to take advantage of each soldier's attributes.

This calculation time problem apparently stems from the fact that each time step simulates a tenth of a second and so for a five second turn, fifty time steps are used. During each step, the computer must determine whether each soldier can "see" any of the enemy troops, complete any orders and determine if any soldier is killed or wounded. All of these calculations require a considerable amount of computation time.

This pace will probably not deter those with a real interest in this type of simulation, but some may find this period of inactivity for the human player somewhat tedious. Since a typical game may require several minutes of simulated time, players can expect to spend several hours on one game. Fortunately SSI has provided players with the ability to save a game in progress on diskette which may be continued at a later time.

Despite the rather lengthy playing time, *Computer Ambush* is a fairly entertaining and probably highly accurate simulation. Although I personally found *Computer Bismarck* a more enjoyable game, fans of that simulation will probably also want to acquire *Computer Ambush.* For more information, contact:

Strategic Simulations, Inc.
450 San Antonio Rd., Suite 62
Palo Alto, CA 94306 □

PROGRAMS FOR THE APPLE II* FROM THE LEADER IN QUALITY SOFTWARE

™

DISK SOFTWARE

*Apple II is a registered trademark of Apple Computer Inc.

APPILOT II™
NEW!! Designed for educators and courseware developers, this high level programming language takes advantage of all the Apple features. Easy to use hi and low-res graphics, Light Pen input, sound and color let you create high-interest interactive lessons.
(48K, $99.95) **I**

ELEMENTARY MATH EDU-DISK™
NEW!! Interactive lessons in elementary addition, subtraction, multiplication and division on 9 skill levels. Extensive color graphics and computer voice maintain student interest. Student scores stored on disk can be accessed only by the teacher.
(48K, $39.95)

THE VOICE™
NEW!! Easily record, edit and playback words or phrases through the Apple speaker. Record your own vocabulary, then add speech to your BASIC programs using Print statements. Guaranteed the best, easiest to use speech software.
(48K, $39.95)

THREE MILE ISLAND™
You are in complete control of a pressurized nuclear reactor! Avoid a melt-down and make a profit in this most realistic simulation. A challenging educational game.
(48K, $39.95)

GLOBAL WAR™
A challenging strategy game for 2 to 9 players. Features a detailed hi-res map of the earth. Players are assigned armies and territories, and then the battle begins. Invade and occupy all territories and you are the winner! Save the current game on disk at any time and restart it later at the same point.
(48K, $24.95) **A**

SUPER-TEXT II™
The professional word processor. An unmatched combination of features make this the most advanced word processor available. Create, edit, merge or print files of any kind—letters, manuscripts, books. Built-in Math functions for preparing financial and summary reports. The most useful program you can own!
(48K, $150.00)

ADDRESS BOOK™
The BEST mailing list program for the Apple! Store 700 addresses per disk. Select by record contents for viewing, label printing or automatic phone dialing. Quickly sort the file in any order.
(48K, $49.95) **A**

MICRO INFORMATION SYSTEM™
Versatile data base system to handle Accounts Payable, Inventory, selective mailings, checking · savings and much more. Easily modify data format to suit your needs.
(48K, $99.95)

U-DRAW II™
It's easy to create hi-res shapes, then rotate, expand or move them with U-Draw II. Layout floor plans, arrange furniture or draw any shapes. Save them on disk and use them in your Basic programs.
(32K, $39.95)

BEST OF MUSE
Five of our most popular cassette games on one disk. Includes Escape, Music Box, Maze Game, Tank War and Side Shows. Hours of entertainment at a bargain price!
(32K, $39.95)

SUPER-LOAD™ CASSETTES

ESCAPE™ (16K, $12.95) **I**

MUSIC BOX™ (8K, $12.95)

APPEN-I™ TEXT EDITOR (8K, $17.95) **I**

TANK WAR™ (16K, $12.95)

MAZE GAME (16K, $12.95) **I**

SIDE SHOWS™ (8K, $12.95) **I**

BASEBALL™ (16K, $14.95) **A**

ELECTRIC CRAYON™ (8K, $17.95)

UNCLE SAM'S JIGSAW™ (32K, $12.95) **A**

GLOBAL WAR™ (32K, $17.95) **A**

U-DRAW™ (16K, $17.95)

For a free catalog or the name of your nearest MUSE Dealer, contact:

MUSE SOFTWARE™

I= Requires Integer Basic
A= Requires Applesoft ROM

330 N. CHARLES STREET
BALTIMORE, MD 21201
(301) 659-7212

CIRCLE 226 ON READER SERVICE CARD

On Effective Documentation

Michael Robinson

Here's a pleasant Saturday-afternoon task: take a program, not a simple one and not one of your own, and write complete documentation for it. The fewer remarks it has before you begin, the better. After you finish your documentation, give it to a novice user and see how well he understands it. Or, read it yourself, and see how well *you* understand it.

I would expect very few people to jump right up and spend their days doing what I just suggested. I said it to make a point: if you for any reason were to hunt for an obscure, complex, poorly-documented program, you would not have to look very far. Hundreds of them exist: programs written by someone who had insufficient regard for the usefulness (and the importance) of good, effective documentation. After the coding and debugging of a program is finished, all too many programmers quit, and unwittingly leave the job half-done. What did they forget? Documentation.

Not all programs need a user manual. Most programs, in fact, do not. But *all* programs need some sort of documentation, some form of explanation showing both how to use the program and how the program works. This explanation may be brief, or it may be verbose. But it must be present, and it must be clear. Even if the program is not to be distributed, it should be documented. Few programmers remember the fine details of their programs for very long after they finish them. But if the need arises to trap an obscure bug, or just to make the program a little bit better, the amount and clarity of documentation provided in initial and subsequent coding efforts will make a great deal of difference in how quickly and how easily the work is done. Everyone needs a reminder, or a re-explanation, now and then. It saves thinking the problem through again. And if a program is to be distributed, then documentation becomes vastly more important. Now, instead of writing for someone already fairly familiar with the code, you must write for someone who has never seen it! It is no easy task.

Michael D. Robinson, Route 4, Box 70, Ringgold, GA 30736.

Possibly the greatest problem in writing effective documentation is the fact that the programmer who created the program often finds himself assuming that the reader knows a great deal more than he actually does. The programmer may find it difficult to explain the program in terms that someone else can easily understand. Being the author, and thus knowing well the background and details of the program's usage and operation, he tends to omit those details from his documentation and assume that the reader already knows and understands them. In fact, he probably does not.

Documentation is not an incidental task, to be done when everything else is finished; it is a basic and very important part of programming.

All documentation is written with the assumption that the reader knows a certain amount of information already. This amount does vary. The Level I Basic manual for the TRS-80, for example, was a strictly tutorial affair, assuming no programming knowledge whatsoever from its reader. The Level II Basic manual for that same computer, on the other hand, assumed that the reader was already familiar with Basic (from the Level I manual, perhaps?), and said so right at the beginning.

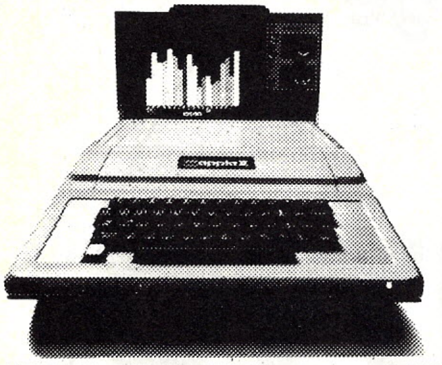
Documentation for personal users must also be written with an understanding of the sequence of use that personal programs undergo. The reader will probably first experiment with the system, becoming familiar with it through the more tutorial segments of the manual, and then begin actually to use the system for his particular task. Later, he will want answers to specific questions, without having to read a bunch of material he has already examined. He will then want to find quick definitions and summarizations — reference material, not tutorial material.

Thus, good documentation must contain all of these: quick-reference summaries, detailed descriptions, and textual discussions of the material. In many documentation manuals, the quick-reference summaries appear at the beginning, with references to more detailed discussions which are presented later in the manual. The experienced user can flip to the summaries and be satisfied, while the less-experienced user can refer to the text discussions.

Besides information for the user of the program, all documentation should contain a description of how the program operates. This information can, of course, be much more technical and less user-oriented than the user documentation, since the programmer to whom the technical documentation is addressed can be assumed to possess skill comparable to that of the original programmer, but the documentation must nevertheless be equally clear. All major program modules must be identified, preferably by remarks contained in the program itself. This is one reason that documentation should always include a listing of the program.

Much documentation consists of comments and annotations in the source code. The programmer can get a good, leisurely look at the program, and easily see from the contained comments what the program as a whole is doing as well as what any given section of the program is doing. Therefore the documentation should also contain an identification and description of all major variables, labels, and procedures in the program, and a concise description of the data base used by the program.

Documentation should identify the name of the program, begin with the programmer's name, the date the program



Documentation, cont'd. . .

was originally written, the current version of the program, and its date. The program should contain this information in header comments, and any manuals should contain the information, as in: "For Version 6.03 of MAILIST, by Ima Good Programmer, written 6/14/79, last updated on 6/22/79 by Ima G. Programmer." (Patches should also be denoted by remarks, including who put the patch in, when it was inserted, and what error the patch was intended to correct — especially since patches are usually last-minute insertions and can be rather difficult to understand.)

The experienced user can flip to the summaries and be satisfied.

Documentation should thus consist of four distinct sections: (1) an introduction, to tell the user in simple, non-technical terms what the program is doing; (2) a quick-reference guide to let the experienced user of the program get answers to questions quickly; (3) a more in-depth explanation of the program's operation and use, intended more for the less-experienced program user; and (4) a technical description of how the program works to allow other programmers to easily service the material.

Documentation is not an incidental task, to be done when everything else is finished; it is a basic and very important part of programming. It makes the difference between a mediocre program and a good one. Documentation is the difference between a job half-done and a job well-done. □


```
10 PRINT " MAY I HELP YOU ?"
20 INPUT N$
RUN
MAY I HELP YOU ?
?
```

We see it as a good way to get things done.

Apple has built a great computer. We at CCS have built a great line of peripherals and components to expand the Apple. To do almost anything you want to get done with a computer.

If you want to do business with an Apple, we've got tools to connect the Apple to standard business printers and terminals. Or to modems, for communications over telephone lines, with other computers, even with other Apples.

If you want to apply your Apple to engineering, scientific, or graphic projects, we've got tools for high-powered, high-speed math functions, and fast, high resolution graphics. And tools to connect the Apple to lab test equipment like function generators or plotters.

And we have tools to connect the Apple to the outside world, including A/D converters and interval timers with external interface.

We make components for the S-100 bus, the PET, and the TRS-80, too. We built our products to deliver hard-nosed value to the OEM, and to the inventor who knows the best, at prices that are unbeaten.

To find out how much computer your Apple II can be, see things our way. Because for serious users with serious uses for the Apple, we've got the tools.

California Computer Systems
250 Caribbean Sunnyvale, CA 94086 (408) 734-5811

CIRCLE 123 ON READER SERVICE CARD

The MAGiC WAND™ is

ALMOST PERFEC.

We've been saying it for a few months now, and the reviewers seem to agree.

"Until I saw the Magic Wand, if I were allowed to own one and only one editor, Word Star* would have been it. . . . My personal preference is for Pencil or Magic Wand for text creation."

Jerry Pournelle
On Computing, Summer 1980

"The basic functions of the Magic Wand editor are as easy to learn as those of Electric Pencil*. . . . Magic Wand dominates in the area of print formatting."

Larry Press
On Computing, Summer 1980

"Of all the word processors I have used (and that includes a dozen or more), the Magic Wand is the most versatile. The Wand has almost all of the features of other processors, plus many new ones of its own. It measures up to even the word-processing software running on the largest mainframe computers."

Rod Hallen
Microcomputing, June 1980

"The Magic Wand is one of the most flexible word processing packages available, and should be considered by any potential word processing purchaser."

Glenn A. Hart
Creative Computing, August 1980

Available for both the CP/M® and OASIS operating systems

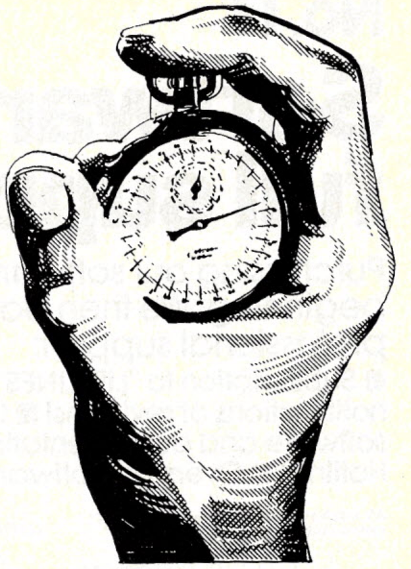
small business applications, inc.

3220 Louisiana • Suite 205 • Houston, Texas 77006 • 713-528-5158

Electric Pencil is a trademark of Michael Shrayer Software, Inc.
WordStar is a trademark of Micro Pro International, Inc.
CP/M is a registered trademark of Digital Research Corp.

Extremely Fast Sorting

Together with a Sidelight on Computer Pseudo-Science



W. D. Maurer

> *"We'll get this thing sorted in 2n steps," said the detective, flashing his Ph.D., "if nobody leaves the premises."*
>
> *("But I can do it a lot faster using this file I've hidden away," thought our hero.)* — *TN*

Sorting, either on big computers or on little ones, is one of the most time-consuming tasks a computer faces. It is no accident that computer scientists through the ages have been searching for faster ways to sort. The problems inherent in this research have recently been complicated by the emergence of a pseudo-science known as "lower-bounds analysis," which attempts to place lower bounds on how fast we can perform certain operations, of which sorting is a prime example.

In this paper there will be described a method which can be used in most production environments to sort n items in a number of steps which is less than 2n. This is apparently in flagrant violation of a dictum which states that the absolute minimum number of steps we can take in a sort is n log n, where the logarithm is to the base 2. Partly in order to explain this discrepancy, we shall re-think the basic assumptions behind the above dictum and others like it, and try to discern how this particular branch of computer science went so badly astray.

In order to understand more clearly the nature of the faulty reasoning which applies here, let us consider an analogy. During the nineteen-sixties there was a lively argument as to whether computers could ever play chess at an expert level. There were serious thinkers on both sides of the argument; as late as 1972, Dreyfus was saying, in his book *What Computers Can't Do,* that: "Chess . . . presents the problem inevitably connected with choice mazes: exponential growth. Alternative paths multiply so rapidly that we cannot even run through all the branching possibilities far enough to form a reliable judgment as to whether a given branch is sufficiently promising to merit further exploration."

W. D. Maurer, George Washington University, S.E.A.S., Washington, DC 20052.

Let us make this argument more specific. Suppose that in a given chess position there are 20 possible moves for each player, on the average. To "look ahead" to a depth of four moves for each player thus requires 25,600,000,000 positions to be analyzed; if each analysis takes 100 microseconds, the total time required is four weeks. The data in the above example are plausible; there are varying estimates of the numbers, but the same conclusion is always reached: the amount of time that would be required to make even one chess move (at expert level) is unreasonably long.

Many computers are not used to their full capacity and any attempt to save computer time makes little sense unless it simultaneously makes programming easier.

Such an argument seems strange today, when expert-level chess programs abound. But what is even stranger is that, from a logical point of view, *there is nothing wrong with the above argument.* The only mistakes are in the *assumptions* from which the argument proceeds to its false conclusion. The key assumption is that each of the 25,600,000,000 positions actually has to be analyzed — that there cannot exist, in other words, techniques for removing from consideration whole classes of these without analysis, so as to reduce the number of analyzed positions to a more manageable level.

This pattern by which false conclusions are derived from false hypotheses by means of perfectly valid (and therefore publishable) arguments is what characterizes much of lower-bounds analysis. Let us consider the argument used to "prove" that sorting n items must take at least n log n steps. According to this argument, if a program has k different possible outcomes, then at least log k steps must be taken. To see this, consider the following. A branch in a program which is the result of a comparison allows the program to have two possible outcomes. If after this branch (whichever way it goes) we make a second comparison, resulting in another branch, we now have four possible outcomes. After a third branch, we have eight possible outcomes, and so on. Thus after j comparisons, we have 2^j possible outcomes, and the above result now follows by setting $k = 2^j$.

How many possible outcomes are there of a sort? Since there are n items to sort, and these can be arranged in n! possible ways, the total number of outcomes is n!. The logarithm of this, to the base 2, happens to be comparable to n log n (I won't bother with the mathematical details of this), and so that becomes the lower bound on the timing of a sort.

And yet I am about to describe a method of sorting that will work in most production environments, and that will take, in a typical situation, no more than 2n steps. How is this possible? As before, there is no logical difficulty in the argument; the problem is with its underlying assumptions. In this case, the key assumption is that every one of the n! outcomes of the sort is actually possible; or, in other words, that there cannot exist a way to rule out whole classes of these and thus to reduce the number of possible outcomes to a more manageable level. In most production environments, this is false and, when one frees one's mind from artificial constraints, quite obviously false.

Consider a typical sort, in which we start with a file F, sort it, and produce a sorted file G. Since this is a *production* environment, the sort will take place periodically — every day, every week, or the like. Instead of a single file F, we have a series of files F_1, F , . . . , which are sorted to yield files G_1, G_2, ... Our problem is

Software with Manual / Manual Alone

New!

☐ **T/MAKER**—Powerful new tool for preparing management reports with tabular data. Makes financial modeling projects easy. Do you want a weekly profitability report? Set up the table and compute. Just change the sales figures for next week and compute. You have a new report! T/MAKER includes a full screen editor for setting up tables which pages left, right, up and down. Compute includes standard arithmetic, percents, exponents, common transcendental functions, averages, maxima, minima, projections, etc. Requires 48K CP/M, CBASIC-2, CRT Terminal with addressable cursor positioning. ....$275/$25

☐ **BSTAM**—Utility to link one computer to another Ⓜ also equipped with BSTAM. Allows file transfers at full data speed (no conversion to hex), with CRC block control check for very reliable error detection and automatic retry. We use it! It's great! Full wildcard expansion to send *. COM, etc. 9600 baud with wire. 300 baud with phone connection. **Both** ends need one. Standard and Ⓜ versions can talk to one another. This software requires a knowledge of assembler language for installation. ....$150/$10

☐ **BSTMS**—Intelligent terminal program for Ⓜ CP/M systems. Permits communication between micros and mainframes. Sends character data files to remote computers under complete control. System can record character data sent from remote computer systems and data banks. Includes programs to EXPAND and COMPRESS binary files for transmission. This software requires a knowledge of assembler language for installation. ....$200/$25

☐ **WHATSIT?***—Interactive data-base system using associative tags to retrieve information by subject. Hashing and random access used for fast response. Requires CBASIC-2 .$175/$25

☐ **SELECTOR III-C2**—Data Base Processor to † create and maintain multi-key data bases. Ⓜ Prints formatted sorted reports with numerical summaries or mailing labels. Comes with sample applications, including Sales Activity, Inventory, Payables, Receivables, Check Register, and Client/Patient Appointments, etc. Requires CBASIC-2. Supplied in source .....$295/$20

☐ **GLECTOR**—General Ledger option to SELECTOR III-C2. Interactive system provides for customized COA. Unique chart of transaction types insure proper double entry bookkeeping. Generates balance sheets, P&L statements and journals. Two year record allows for statement of changes in financial position report. Supplied in source. Requires SELECTOR III-C2. CBASIC-2 and 56K system. ....$350/$25

DMA

☐ **CBS**—Configurable Business System is a † comprehensive set of programs for defining custom data files and application systems without using a programming language such as BASIC, FORTRAN, etc. Multiple key fields for each data file are supported. Set-up program customizes system to user's CRT and printer. Provides fast and easy interactive data entry and retrieval **with transaction processing**. Report generator program does complex calculations with stored and derived data, record selection with multiple criteria, and custom formats. Sample inventory and mailing list systems included. **No support language required** ....$395/$40

MICROPRO

☐ **SUPER-SORT I**—Sort, merge, extract utility as Ⓛ absolute executable program or linkable module in Microsoft format. Sorts fixed or variable records with data in binary, BCD, Packed Decimal, EBCDIC, ASCII, floating & fixed point, exponential, field justified, etc. Even variable number of fields per record! ....$225/$25

☐ **SUPER-SORT II**—Above available as abso- Ⓛ lute program only ....$175/$25

☐ **SUPER-SORT III**—As II without SELECT/ Ⓛ EXCLUDE ....$125/$25

☐ **DATASTAR**—Professional forms control entry Ⓛ and display system for key-to-disk data capture. Menu driven with built-in learning aids. Input field verification by length, mask, attribute (i.e. upper case, lower case, numeric, auto-dup, etc.). Built-in arithmetic capabilities using keyed data, constant and derived values. Visual feedback for ease of forms design. Files compatible with CP/M-MP/M supported languages. Requires 32K CP/M ....$350/$35

☐ **WORD-STAR**—Menu driven visual word pro- Ⓛ cessing system for use with standard terminals. Text formatting performed on screen. Facilities for text paginate, page number, justify, center and underscore. User can print one document while simultaneously editing a second. Edit facilities include global search and replace, Read/Write to other text files, block move, etc. Requires CRT terminal with addressable cursor positioning ....$445/$40

☐ **WORD-STAR-MAIL-MERGE**—As above with Ⓛ option for production mailing of personalized documents with mail lists from DATASTAR or NAD ....$575/$40

☐ **WORD-MASTER** Text Editor—In one mode Ⓛ has superset of CP/M's ED commands including global searching and replacing, forwards and backwards in file in video mode, provides full screen editor for users with serial addressable-cursor terminal ....$145/$25

Software with Manual / Manual Alone

New!

☐ **MAGIC WAND***—Word processing system with simple, easy to use full screen text editor and powerful print processor. Editor has all standard editing functions including text insert and delete, global search and replace, block move and library files for boiler plate text. Print processor formatting commands include automatic margins, pagination, headings & footings, centered and justified text. Also prints with true proportional spacing, merges with data files for automatic form letters, and performs run-time conditional testing for varied output. Requires 32K CP/M and CRT terminal with addressable cursor. ....$395/$40

☐ **TEXTWRITER III**—Text formatter to justify and Ⓜ paginate letters and other documents. Special features include insertion of text during execution from other disk files or console, permitting recipe documents to be created from linked fragments on other files. Has facilities for sorted index, table of contents and footnote insertions. Ideal for contracts, manuals, etc. Now compatible with Electric Pencil* and Word-Star prepared files ....$125/$20

New!

☐ **DATEBOOK**—Program to manage time just like an office appointment book but using the speed and memory of a computer. Keeps track of three appointment schedules (three dental chairs, three attorneys, etc.) at once. Appointments consist of name, reason for the appointment, the date and time, and the length of the appointment. System can be quickly customized for the individual user. Many helpful features for making, changing, finding, and reporting appointments. Requires 48K CP/M and 180K bytes diskette storage. Not available for Apple CP/M. ....$295/$25

New lower prices for application Software

PEACHTREE SOFTWARE*

☐ General accounting software for small busi- Ⓛ nesses. Each product can be used alone or with † automatic posting to the general ledger. Supplied in source for Microsoft BASIC 4.51.

GENERAL LEDGER	$530/$40
ACCOUNTS PAYABLE	$530/$40
ACCOUNTS RECEIVABLE	$530/$40
PAYROLL	$530/$40
INVENTORY	$660/$40

☐ Other application products supplied in source Ⓛ for Microsoft BASIC 4.51.

† MAILING ADDRESS	$530/$40
PROPERTY MANAGEMENT	$925/$40

GRAHAM-DORIAN SOFTWARE SYSTEMS

☐ Comprehensive accounting software written in Ⓛ CBASIC-2 and supplied in source code. Each Ⓜ software package can be used as a stand- † alone system or integrated with the General Ledger for automatic posting to ledger accounts. Requires CBASIC-2.

GENERAL LEDGER	$805/$40
ACCOUNTS PAYABLE	$805/$40
ACCOUNTS RECEIVABLE	$805/$40
INVENTORY SYSTEM	$555/$40
JOB COSTING	$805/$40
APARTMENT MANAGEMENT	$805/$40
CASH REGISTER	$805/$40

☐ **POSTMASTER**—A comprehensive package Ⓜ for mail list maintenance that is completely menu driven. Features include keyed record extraction and label production. A form letter program is included which provides neat letters on single sheet or continuous forms. Includes NAD file translator. Requires CBASIC-2. ....$150/$20

STRUCTURED SYSTEMS GROUP

☐ Complete interactive accounting software for † business. Each product can be used standalone or with automatic posting to the general ledger. Each product is thoroughly tested and very well documented. Each product requires CBASIC-2.

GENERAL LEDGER	$820/$40
ACCOUNTS RECEIVABLE	$820/$40
ACCOUNTS PAYABLE	$820/$40
PAYROLL	$820/$40
INVENTORY CONTROL	$820/$40

LIFELINES
NEWSLETTER FROM LIFEBOAT

LIFELINES is the first step in software support for the serious microcomputer user. Each issue reports new revisions together with information on the purpose for each such release, be it for correction of "bugs" or the addition of features and facilities.

Feature Articles ! New Software ! Product Comparisons ! Info on CP/M Users Group !

SUBSCRIPTION INFORMATION:

$18 for twelve issues: U.S., Canada, and Mexico.

$40 for twelve issues: all other countries.

$2.50 for each back issue: U.S., Canada, and Mexico.

$3.60 for each back issue: all other countries. Send Check to LIFELINES, 1651 Third Avenue, New York, N.Y. 10028 or use your VISA or MASTERCARD—call (212) 722-1700

Software with Manual / Manual Alone

☐ **ANALYST**—Customized data entry and report- † ing system. User specifies up to 75 data items per record. Interactive data entry, retrieval, and update facility makes information management easy. Sophisticated report generator provides customized reports using selected records with multiple level breakpoints for summarization. Requires a disk sort utility such as QSORT, SUPER-SORT or VSORT and CBASIC-2 ....$250/$15

☐ **LETTERIGHT**—Program to create, edit and type letters or other documents. Has facilities to enter, display, delete and move text, with good video screen presentation. Designed to integrate with NAD for form letter mailings. Requires CBASIC-2 ....$200/$25

☐ **NAD** Name and Address selection system—Interactive mail list creation and maintenance program with output as full reports with reference data or restricted information for mail labels. Transfer system for extraction and transfer of selected records to create new files. Requires CBASIC-2 ....$100/$20

☐ **QSORT**—Fast sort/merge program for files with fixed record length, variable field length information. Up to five ascending or descending keys. Full back-up of input files created ....$100/$20

★ ★ ★ ★ ★ ★ ★
CONDIMENTS
★ ★ ★ ★ ★ ★ ★

☐ **HEAD CLEANING DISKETTE**—Cleans the drive Read/Write head in 30 seconds. Diskette absorbs loose oxide particles, fingerprints, and other foreign particles that might hinder the performance of the drive head. Lasts at least 3 months with daily use. Specify 5" or 8".
Single sided ....$20 each/$55 for 3
Double sided ....$25 each/$65 for 3

☐ **DC 300 Data Cartridges** Specify 450'XL or 300' certified. Pack of 5. ....$100

☐ **FLIPPY DISK KIT**—Template and instructions to modify single sided 5¼" diskettes for use of second side in single sided drives $12.50

☐ **FLOPPY SAVER**—Protection for center holes for 5" and 8" floppy disks. Only 1 needed per diskette. Kit contains centering post, pressure tool and tough 7 mil mylar reinforcing rings for 25 diskettes.

5", Kit	$14.95
5", Rings only	$7.95
8", Kit	$16.95
8", Rings only	$8.95

☐ **PASCAL USER MANUAL AND REPORT**—By Jensen and Wirth. The standard textbook on the language. Recommended for use by Pascal/Z, Pascal/M and Pascal/MT users $12

☐ **THE C PROGRAMMING LANGUAGE**—By Kernighan and Ritchie. The standard textbook on the language. Recommended for use by BDS C, tiny C, and Whitesmiths C users . .$12

☐ **STRUCTURED MICROPROCESSOR PROGRAMMING**—By the authors of SMAL/80. Covers structured programming, the 8080/8085 instruction set and the SMAL/80 language $20

☐ **ACCOUNTS PAYABLE & ACCOUNTS RECEIVABLE—CBASIC**—By Osborne/McGraw-Hill $20

☐ **GENERAL LEDGER—CBASIC**—By Osborne/McGraw-Hill $20

☐ **PAYROLL WITH COST ACCOUNTING—CBASIC**—by Osborne/McGraw-Hill $20

★ ★ ★ ★ ★ ★ ★
Hearty Appetite.
★ ★ ★ ★ ★ ★ ★ ★

*CP/M and MP/M are trademarks of Digital Research.
Z80 is a trademark of Zilog, Inc.
UNIX is a trademark of Bell Laboratories.
WHATSIT? is a trademark of Computer Headware.
Electric Pencil is a trademark of Michael Shrayer Software.
TRS-80 is a trademark of Tandy Corp.
Pascal/M is a trademark of Sorcim.
SoftCard is a trademark of Microsoft.
Apple is a trademark of Apple Computer.
PASM, PLINK, BUG and μBUG are trademarks of Phoenix Software Associates Ltd.
CPAids is a trademark of Computer Tax Service, Inc.
MAGIC WAND is a trademark of Small Business Application, Inc.
Peachtree Software is a trademark of Retail Sciences, Inc.

† Recommended system configuration consists of 48K CP/M, 2 full size disk drives, 24 x 80 CRT and 132 column printer.

Ⓜ Modified version available for use with CP/M as implemented on Heath and TRS-80 Model I computers.

Ⓛ User license agreement for this product must be signed and returned to Lifeboat Associates before shipment may be made.

Ⓘ This product Includes/eXcludes the language Ⓧ manual recommended in Condiments.

Ⓢ Serial number of CP/M system must be supplied with orders.

Ⓩ Requires Z80 CPU.

Ordering Information

MEDIA FORMAT ORDERING CODES
When ordering, please specify format code.

LIFEBOAT ASSOCIATES MEDIA FORMATS LIST. Diskette, cartridge disk and cartridge tape format codes to be specified when ordering software for listed computer or disk systems. All software products have specific requirements in terms of hardware or software support, such as MPU type, memory size, support operating system or language.

Computer system	Format Code
Altair 8800 Disk	See MITS 3200
Altos	A1*
Apple + SoftCard 13 Sector	RG
Apple + SoftCard 16 Sector	RR
AVL Eagle	RB
BASF System 7100	RD
Blackhawk Single Density	Q3
Blackhawk Micropolis Mod II	Q2
CDS Versatile 3B	Q1
CDS Versatile 4	Q2
COMPAL-80	Q2
Cromemco System 3	A1*
Cromemco Z2D	R6
CSSN BACKUP (tape)	T1 #
Delta	A1*
Digi-Log Microterm II	RD
Digital Microsystems	A1*
Discus	See Morrow Discus
Durango F-85	RL
Dynabyte DB8/2	R1
Dynabyte DB8/4	A1*
Exidy Sorcerer + Lifeboat CP/M	Q2
Exidy Sorcerer + Exidy CP/M	Q4
Heath H8 + H17/H27	P4
Heath H89 + Lifeboat CP/M	P4
Heath H89 + Magnolia CP/M	P7
Helios II	See Processor Technology
Horizon	See North Star
iCOM 2411 Micro Floppy	R3
iCOM 3712	A1
iCOM 3812	A1*

Computer system	Format Code
iCOM 4511 5440 Cartridge CP/M 1.4	D1 #
iCOM 4511 5440 Cartridge CP/M 2.2	D2 #
IMS 5000	RA
IMS 8000	A1*
IMSAI VDP-40	R4**
IMSAI VDP-42	R4**
IMSAI VDP-44	R5**
IMSAI VDP-80	A1**
Intecolor	See ISC Intecolor
Intel MDS Single Density	A2
Intel MDS Double Density	A5
Intertec SuperBrain DOS 0.1	R7
Intertec SuperBrain DOS 0.5-2.X	RJ
Intertec SuperBrain DOS 3.X	RK
ISC Intecolor 8063/8360/8963	A1
Kontron PSI-80	RF
Meca 5¼"	P6
Micromation (Except TRS-80 below)	A1*
Micropolis Mod I	Q1
Micropolis Mod II	Q2
MITS 3200/3202	B1
Morrow Discus	A1*
Mostek	A1
MSD 5¼"	RC
North Star Single Density	P1
North Star Double/Quad	P2
Nylac Single Density	Q3
Nylac Micropolis Mod. II	Q2
Ohio Scientific C3	A3
Onyx C8001	T2 #
Pertec PCC 2000	A1*
Processor Technology Helios II	B2
Quay 500	RO
Quay 520	RP
RAIR Single Density	R9

Computer system	Format Code
RAIR Double Density	RE
Research Machines 8"	A1
Research Machines 5¼"	RH
REX	Q3
Sanco 7000 5¼"	RQ
SD Systems 8"	A1*
SD Systems 5¼"	R3
Sorcerer	See Exidy Sorcerer
Spacebyte	A1
SuperBrain	See Intertec
Tarbell	A1*
TEI 5¼"	R3
TEI 8"	A1*
Thinkertoys	See Morrow Discus
TRS-80 Model I 5¼"	R2
TRS-80 Model I + FEC Freedom	RN
TRS-80 Model I + Micromation	A4*
TRS-80 Model I + Omikron 5¼"	RM
TRS-80 Model I + Omikron 8"	A1
TRS-80 Model I + Shuffleboard 8"	A1
TRS-80 Model II	A1*
VDP-40/42/44/80	See IMSAI
Vector Graphic	Q2
Vector MZ	Q2
Versatile	See CDS Versatile
Vista V80 5¼" Single Density	P5
Vista V200 5¼" Double Density	P6
Zenith Z89 + Lifeboat CP/M	P4
Zenith Z89 + Magnolia CP/M	P7

Prices reflect distribution on 8" single density diskettes. If a format is requested which requires additional diskettes, a surcharge of $8. per additional diskette will be added.

Prices F.O.B. New York. Shipping, handling and C.O.D. charges extra.

Manual cost applicable against price of subsequent software purchase.

The sale of each proprietary software package conveys a license for use on one system only.

* Single-Side Single-Density disks are supplied for use with Double-Density and Double-Side 8" soft sector format systems.

** IMSAI formats are single density with directory offset of zero.

\# A media surcharge of $25 for orders on tape formats T1 and T2 and of $100 for orders on disk formats D1 and D2 will be added.

The list of available formats is subject to change without notice. In case of uncertainty, call to confirm the format code for any particular equipment.

(212) 860-0300

Lifeboat Associates
1651 Third Avenue, N.Y., N.Y. 10028

Sorting, cont'd...

then to sort a typical file F_i to produce the corresponding sorted file G_i. Also since this is a production environment, there are backup files in existence, namely F_{i-1} and G_{i-1}. Furthermore there is some relationship between F_i and F_{i-1}; we shall assume that F_i was derived from F_{i-1} by certain transactions (insert, delete, replace) and that the total number of these transactions is small in comparison to the total size of F_i. This last assumption is often, but not always, valid, which is why we have specified that our method will work in most (but not all) production environments.

The first step in our sorting method is to set up still another file, which we call T_i. We assume that F_i is derived from F_{i-1}, not all at once, but as a result of a series of transactions which took place since F_{i-1} was sorted to yield G_{i-1}. Every time such a transaction takes place, in addition to updating F_{i-1}, we record the transaction, together with its date, on the file T_i. The extra time this takes is proportional to the number of transactions, which we shall call m.

To perform the sort, we first sort the file T_i, using the date as a secondary key. That is, if there are transactions on T_i with identical keys, these are arranged in order of their dates (which are recorded on T_i). This is a conventional sort and takes m log m steps. We are assuming that m is small compared with n, and we will make this explicit by requiring that m log m be less than n. For example, if there are 1000 records in the file and 100 transactions (n = 1000, m = 100), then the condition is satisfied, since m log m is approximately 700 in this case. Thus the number of steps in this initial sort will be less than n; we denote the resulting sorted file by U_i.

Finally, we produce the new sorted file G_i by merging U_i with the backup file G_{i-1}. Our reasoning is as follows. Every record in F_i is also in G_i; every record in F_{i-1} is also in G_{i-1}. Hence the only differences between G_{i-1} (which we have kept around) and G_i (which we are to produce) are those indicated by the transactions which took place on F_{i-1} to produce F_i. Each of these transactions, in turn, is on T_i; and whenever we have several transactions on the same record in G_i, we have them in their historical order, so that their performance always leads to the correct final result.

The importance of this last point may be illustrated as follows. Suppose that F_i is a personnel file, normally sorted by employee's name, which is sorted every month by zip code in order to mail paychecks. Suppose that at the beginning of the month one employee moves from the main office M to a branch office B and then moves back before the end of the month. Thus there are two transactions involving the "which-office?" field: "change to B" and "change to M." These have to be done in the correct order; otherwise, the final value of the "which-office?" field will be B rather than M. Using the date as a secondary key guarantees to us that the correct order will be used.

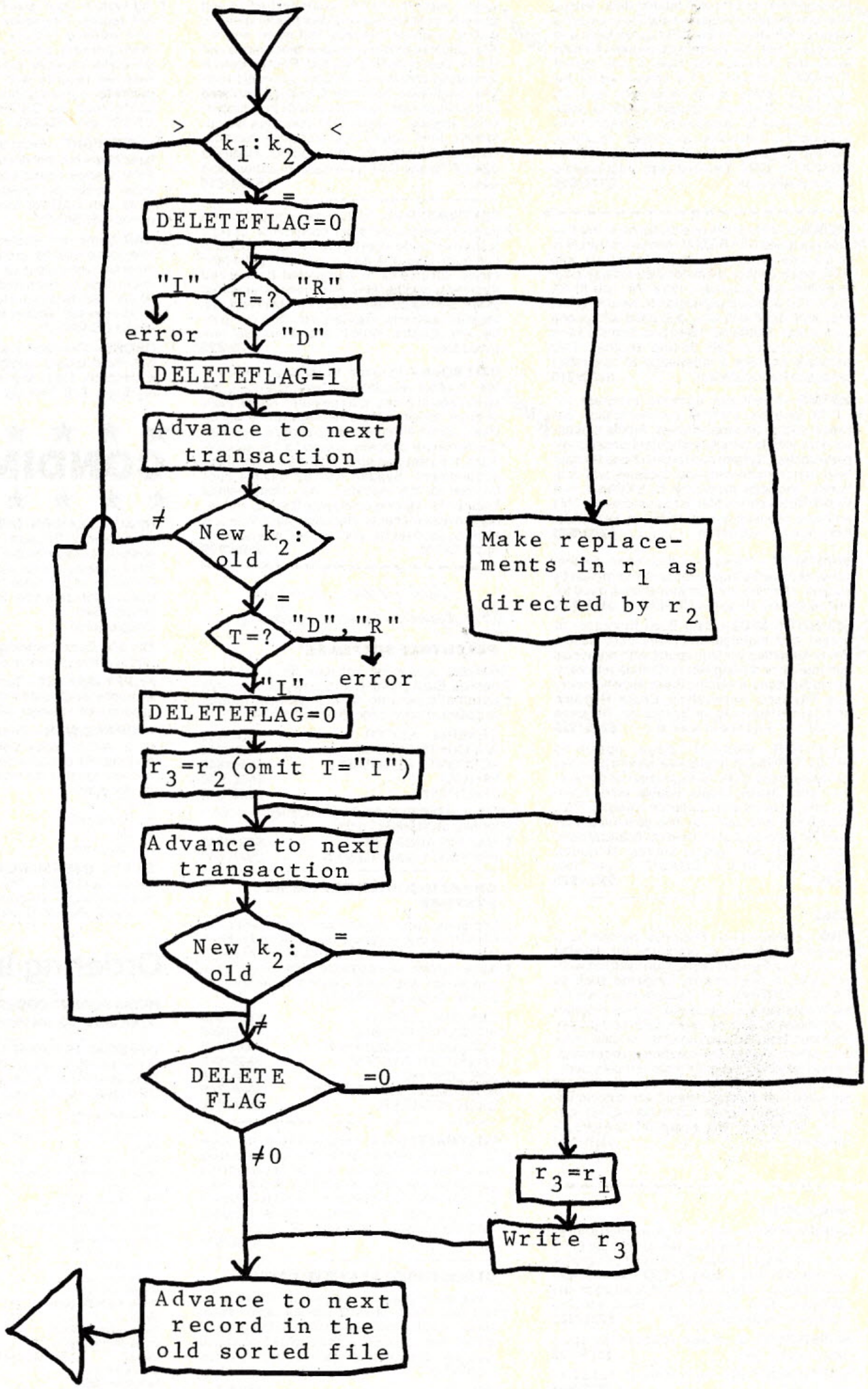


Figure 1

The reason for sorting T_i is that the sorted file U_i can now be merged with G_{i-1} in n steps, so that the total number of steps is less than 2n, as we specified earlier. (A merge of two files, producing a result file of length n, and taking n steps, works only if both files are sorted, and in the same way.) The merge must actually be modified to produce, on the result file G_i, the *result of applying* the transactions in U_i, rather than the transactions themselves. For example, a record in F_{i-1} might be of the form

N="SMITH, JOHN J.",S=2400000, Z=20037,W="M", . . . (*)

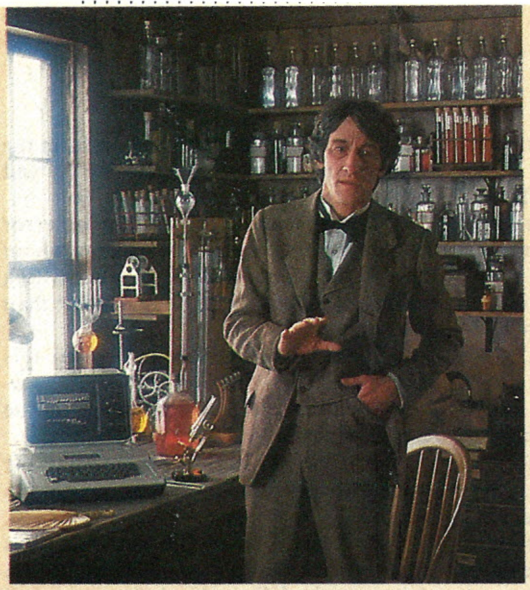
If Smith moves from the main office to the branch office, this record would be changed to read

N="SMITH, JOHN J.",S=2400000, Z=20037,W="B", . . . (**)

and, at the same time, a transaction record would be written onto T_i. Assuming that the sort is on the field Z (the zip code), the transaction record would be of the form

T="R",Z=210037,W="B" (***)

Here T="R" stands for "transaction type = replace"; we are performing a replacement



Sorting, cont'd...

in the record whose key (when it gets into G_{i-1}) will be Z=20037, and the field that is being replaced is the W field, which is being changed to "B." This record (***) will also appear in U_i, since U_i was produced by sorting T_i, while the original record (*) will appear in G_{i-1}, since this is a sorted version of F_{i-1}. During the merge, we are reading both files in sorted order, and thus we will come to the original record (*) and the transaction record (***) at the same time. Instead of putting both of these on the output file, which is what would happen in a normal merge, we *apply* (***) to (*), producing (**), which is written on the output file. Specifically, in this case, we would change the W="M" to W="B."

In order to explain the workings of multiple replacements to a record, as well as "transaction type = insert" and "transaction type = delete," we review the basic principles of merging. In each of our three files (G_{i-1}, U_i, and G_i, in this case) we have a *current record;* we denote these three current records by r_1, r_2, and r_3, respectively. These have keys k_1, k_2, and k_3, respectively, the key being the zip code in this case. In each file we also have an *initial segment* (the part of the file that precedes the current record); the initial segments of G_{i-1} and U_i have already been merged to form the initial segments of G_i. In a normal merge, we would proceed as follows:

1. If $k_1 < k_2$, then we set $r_3 = r_1$ and advance the current record of the first file by 1.

2. If $k_1 > k_2$, then we set $r_3 = r_2$ and advance the current record of the second file by 1.

3. If $k_1 = k_2$, then we do either of the above (if duplicate keys are allowed in the result file); or we do both of the above (if they are not).

4. In any case, we then advance the current record of the third file by 1.

Our modified merge will proceed as follows. Remember that r_1 is the current record in the old sorted file, while r_2 is the current record in the transaction file.

a. If $k_1 < k_2$, then there is no transaction which applies to the current record; we proceed as in 1. above.

In b. and c. below, let us initially assume that r_2 is the only record in the transaction file whose key is k_2.

b. If $k_1 > k_2$, then r_2 does not apply to r_1, and therefore r_2 must be of "transaction type = insert" (otherwise we have an error). For this transaction type, *all* the fields in r_3 are specified in r_2, and we proceed as in 2. above (except that the special field T="I," denotes "transaction type = insert," is omitted in r_3).

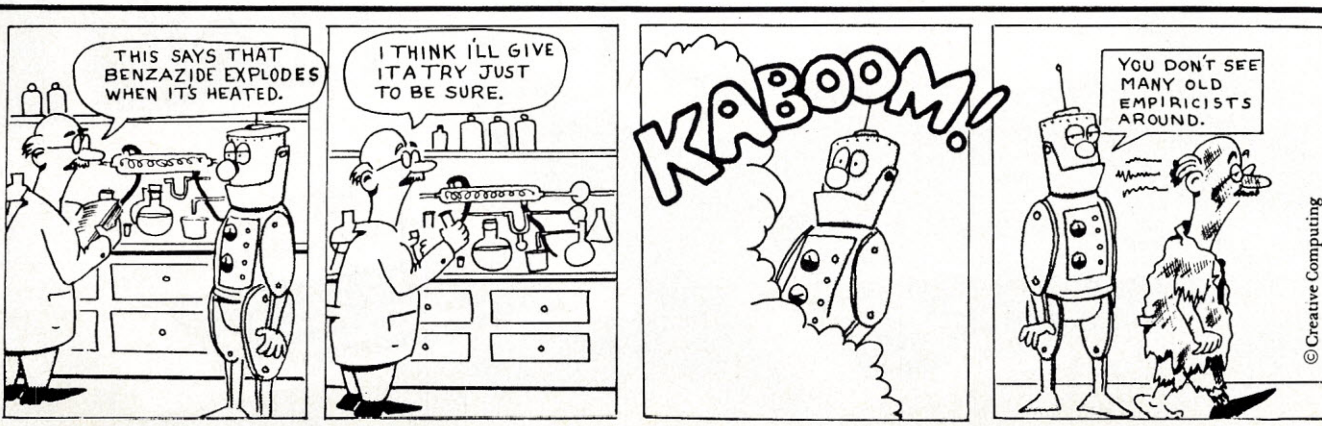
c. If $k_1 = k_2$, then r_2 applies to r_1, and there are two subcases. If r^2 is of "transaction type = replace," then we perform replacements in r_1 as directed by r_2; set r_3 to the result; and advance the current record of the third file by 1. If r_2 is of "transaction type = delete," then we do *not* advance the current record of the third file by 1, since r_1 is to be deleted. In either case, we advance the current records of *both* the first and the second files by 1.

This pattern by which false conclusions are derived from false hypotheses by means of perfectly valid (and therefore publishable) arguments characterizes much of lower-bounds analysis.

If there are several records in the transaction file with key k_2, then these will occur in historical order. In this order, a replace cannot be followed by an insert; a delete can be followed *only* by an insert; and an insert cannot be followed by another insert. The final disposition of r_1 may be as in b. or c. above (either subcase), depending on whether the *last* of the records with key k_2 is of type "insert," "replace," or "delete." The required logic is illustrated by the flowchart of Figure 1; the steps are as follows.

1. If $k_1 > k_2$, proceed to step 9; if k_2, proceed to step 14.
2. We have $k_1 = k_2$ (case c. above). Set DELETEFLAG = 0.
3. If r_2 is a replace, then perform the replacement in r_1 as directed by r_2, and proceed to step 11.
4. If r_2 is an insert, we have an error (we are trying to insert a new record whose key is the same as that of an existing record; we should be replacing rather than inserting).
5. Set DELETEFLAG = 1. (Here r_2 is a delete.)
6. Advance the current record in the transaction file by 1.
7. If the key in the new record of the transaction file is unequal to k_2, proceed to step 13.
8. If the type of the new record is anything but "insert," we have an error. (A delete can be followed only by an insert.)
9. The type of the current record is "insert." Set DELETEFLAG = 0.
10. Set $r_3 = r_2$ (with the special field T="I," denoting "transaction type = insert," omitted).
11. Advance the current record in the transaction file by 1.
12. If the key in the new record of the transaction file is equal to k_2, go back to step 3.
13. If DELETEFLAG $\neq$ 0, go to step 16. (This tells us that the last record whose key is k_2 was a delete, so that r_1 is to be deleted and a new record is not to be written on the result file.)
14. Copy r_1 into r_3.
15. Write r_3, advancing the result file.
16. Advance the current record in the old sorted file by 1.

In closing, we must note two cautions about our sorting method; despite the fact that it is fast, it is not universally recommended. In the first place, many computers (small systems more often than large ones) are not used to their full capacity. In such a setting, *any* attempt to save computer time makes little sense unless it simultaneously makes programming easier. Also *every* change to F_{i-1} must be accompanied by writing a new record on the transaction file. The importance of this point is that it is quite common for a commercial programming system to be written by several programmers, so that it becomes difficult to identify every place in the system at which a change to F_{i-1} is made. □



PEARL™!

Tranquility and Power

PEARL writes programs. PEARL offers creative programming beyond the confines of the way you've done it before.

PEARL leads you to the quiet of centered creative space. PEARL is a program which writes programs in BASIC and is therefore called an application generator. PEARL frees you from the routine work, letting you create new programs within the tranquility and power of freedom. PEARL becomes a powerful extension of yourself and writes the routine but vitally important portions of every application. Creativity becomes your new way.

Just as a BASIC compiler frees you from the many decisions of machine language (how to represent strings and numbers, I/O details, etc.), PEARL frees you from the many decisions required when writing programs in BASIC (designing overall logic flow, menus, details of screen interaction, etc.). For programmers, PEARL Levels 2 and 3 provide unlimited creative range. For the nonprogrammer, PEARL Level 1 allows a calm, easy approach to generating new software.

Let PEARL's power free you to focus on the creative aspects of programming and system design. PEARL 1 ($130), PEARL 2 ($350) and PEARL 3 ($650) require CP/M* and CBASIC**. PEARL 3 also requires QSORT***. Call Ray Stow. Dealer inquiries invited.

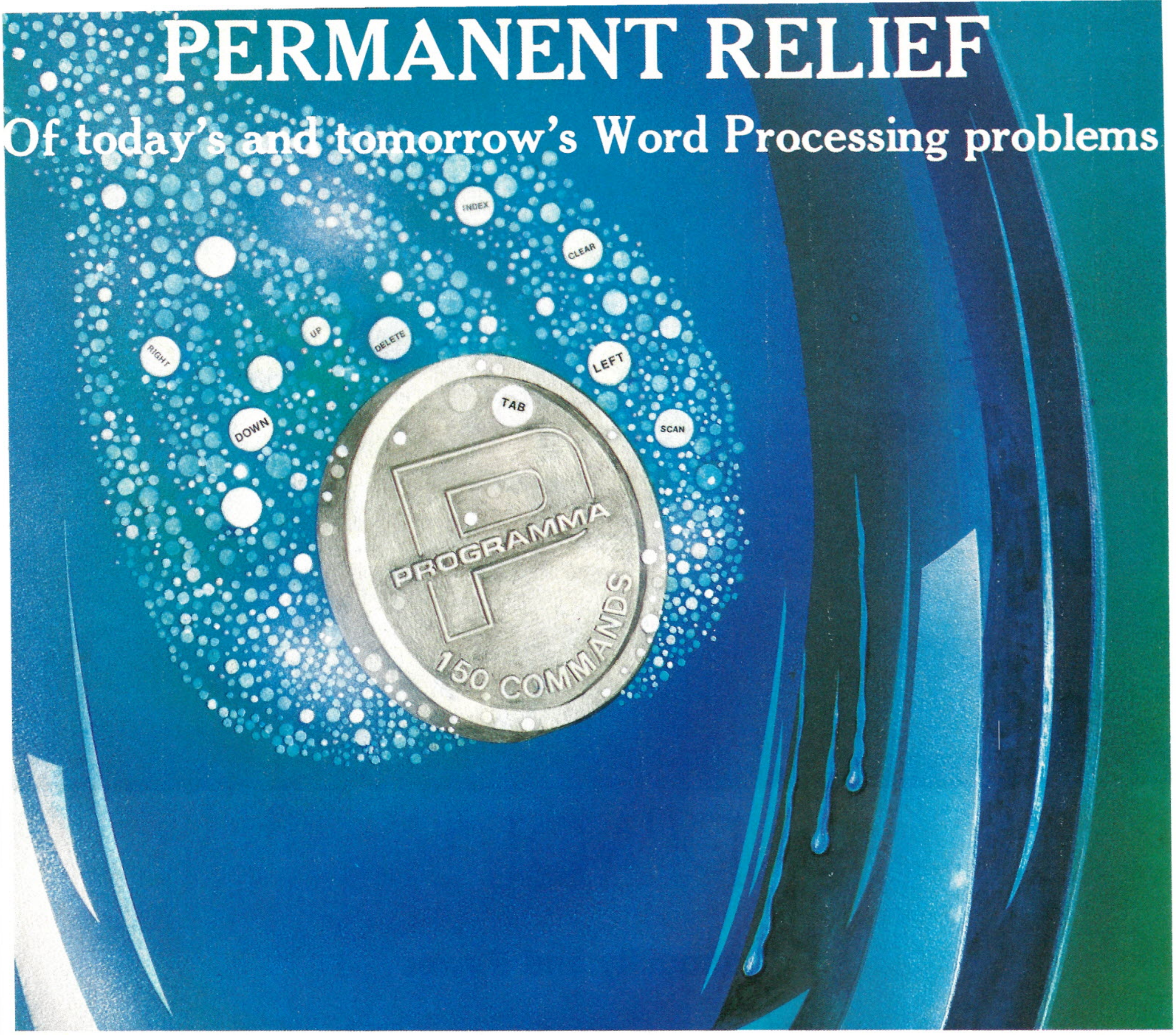
evolutionary software for the 1980's

CPU International • 503-370-8653
P.O. Box 12892 • Salem, OR 97301

CIRCLE 134 ON READER SERVICE CARD

*CP/M is a trademark of Digital Research
**CBASIC is a trademark of Software Systems
***QSORT is a trademark of Structured Systems Group

PERMANENT RELIEF
Of today's and tomorrow's Word Processing problems



Apple PIE + Formatter

Apple PIE (Programma International Editor) nd FORMAT (text formatter) offer full strength olutions to today's word processing problems. 'hese versatile, powerful programs provide locument preparation and word processing apabilities previously found only on much larger omputer systems.

PIE is a general purpose, full screen editor ıat uses control keys and function buttons to rovide a full range of editing capabilities such as earch and replace, delete, copy, insert, move. hanges may be made directly anywhere on the reen and are shown as they are performed.

FORMAT uses simple instructions nbedded in the input text to describe the desired ppearance of the final document. It handles entering, underlining, indenting, page numbering, margins, headers, footers, even form letters, and includes a proofing capability.

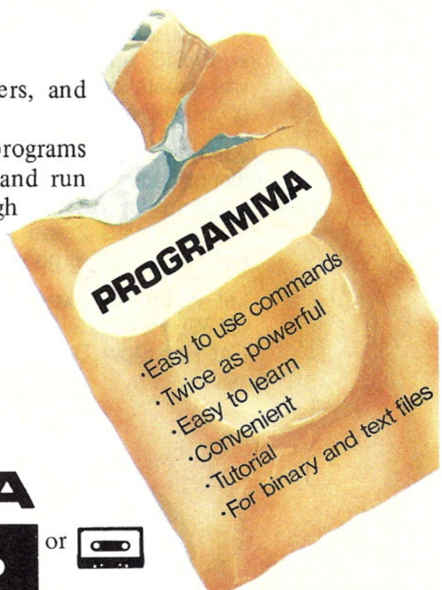
These high-quality, cost-effective programs come with comprehensive documentation and run on a 32K Apple II. They are available through your local computer store or direct from Programma International, Inc. at the introductory price of $79.95*.

VIDEX VERSION T.M.
DOUBLE VISION T.M.
SUPR TERM VERSION T.M.
STANDARD VERSION

*December 1, $129.95.

PROGRAMMA
3400 Wilshire Boulevard
Los Angeles, California 90010

or



Simple enough for the beginner. Versatile enough for the professional.

CIRCLE 230 ON READER SERVICE CARD

Systems Analysis and Small Computers

William R. Feeney

If your involvement with computers has an exact and specifiable goal, this article may help you with it. If your interests are diffuse and evolving, then it may be good for a laugh or two.

How do *you* proceed in a project? Do your projects look more like the left flowchart (Figure 1) or like the right one? If your approach to a project tends to be haphazard, read on!

Commercial Computer Projects

Systems analysts who design and build complex computer systems for businesses and for other applications have been thinking about procedures for decades. Their solutions are particularly good, since these very same solutions can be applied to the less demanding projects of the casual computer expert. [*less demanding? — Ed.*]

Two aspects of commercial computer projects are important here, namely the sequencing of the tasks and the role of documentation in a systems project.

Designing computer systems is complex and involves endless detail. Project personnel have discovered that tasks are accomplished more efficiently and with less confusion when a systematic approach is used. The favored approach is a common sense ordering of tasks arranged into steps or phases of a systems analysis.

Systems analysis, as it is used by commercial data processing projects, usually includes five phases:

1. *Feasibility Study:* This is a quick look at the project to determine if it is worth doing.

2. *Analysis Phase:* Here the project team examines the business context of the project and performance expectations are decided upon. When written up, these are called Performance Specifications.

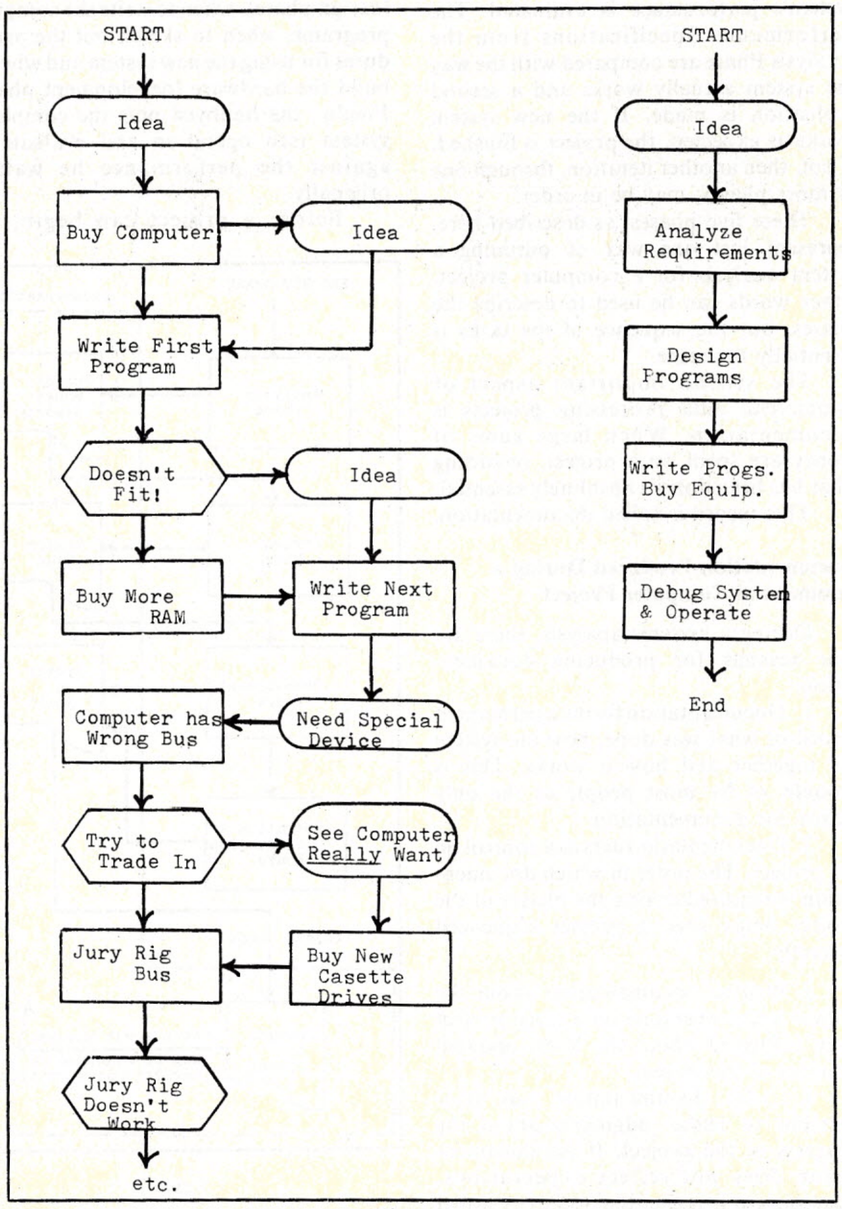


Figure 1.
Which pattern do your projects follow?

William R. Feeney, School of Business Administration
San Diego State University, San Diego, CA 92182.

Systems, cont'd. . .

3. *Design Phase:* The proposed system is divided into separate components. Computer programs and procedures for the users of the new system are developed in this phase. Sometimes a hardware system is also designed, but frequently the newly designed programs can be implemented on an existing computer system. The Design Specifications are written at this time.

4. *Development Phase:* The project team produces the coding for the computer programs and writes up the detailed procedures. When necessary, computer hardware systems are assembled, installed and checked out. The final set of specifications, called System Specifications, are written at this time.

5. *Operation and Evaluation Phase:* The project team gets the software, people procedures and hardware working in this phase. After an initial break-in period, the system's performance is evaluated. The Performance Specifications from the Analysis Phase are compared with the way the system actually works and a second evaluation is made. If the new system works as expected, the project is finished. If not, then another iteration, through one or more phases, may be in order.

These five phases, as described here, represent just one way of outlining a system analysis for a computer project. Other words can be used to describe the phases, but the sequence of the tasks is essentially the same.

The second important aspect of commercial data processing projects is documentation. When large sums of money are spent on a project, recording what has been done is absolutely essential. This record is called documentation.

Documentation Produced During a Commercial Computer Project

During a systems analysis, there are four reasons for producing documentation:

1. Documentation forms a permanent record of what was done, how the system fits together and how it works. This is thought of by most people as the only reason for documentation.

2. Documentation acts as a control on the project. The order in which documentation is required during the phases of the project helps keep a project sequenced correctly.

3. Written documentation about the project keeps everyone on a large project team informed about the latest updated versions of the system.

4. Lastly, documentation aids management to make judgments about the progress of the project. If the documentation does not reflect expectations, management can alter the project or stop it altogether.

The computer hobbyist obviously does not have to keep a large project team informed of his progress; he or she is usually the entire project team! If any others are involved in the effort, conversations are usually sufficient to inform them of the latest developments. Also, the hobbyist does not need to get management approval to continue his project; he is management, giving himself the "OK" to proceed.

What the hobbyist often needs, though, is guidance in planning the project and some advice on what to write down about the project for future reference.

System Analysis Phases for the Hobbyist

At least four of the phases used in commercial data processing projects need to be used by the hobbyist. These four phases will guide the hobbyist in deciding the following: how to design his system's software, hardware and procedures (design phase), when to write the computer programs, when to sketch out the procedures for using the new system and when to build the hardware (development phase). Finally, the hobbyist puts the completed system into operation and evaluates it against the performance he wanted originally.

Before a project can begin, the hobbyist must have a general notion of his purpose. This could be as general as wanting to buy a computer to play electronic games or as detailed as using a computer to control every aspect of security for his household.

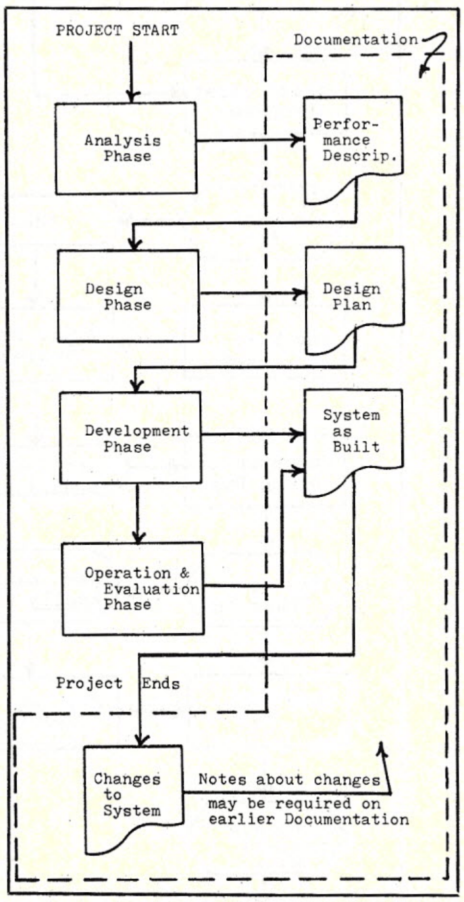


Figure 2.
The Phases and Documentation of a Systems Project

Here again are the phases, with a more detailed description. Figure 2 shows them and their accompanying documentation.

These five phases, as described here, represent just one way of outlining a system analysis for a computer project. Other words can be used to describe the phases . . .

1. *Analysis Phase* — The hobbyist analyzes the *constraints* acting on him and his idea for a new system. How much time and money does he have? How knowledgeable is he about getting a system running? What tools and work space are available to him? Once the hobbyist has an accurate idea of his constraints, he can correct any deficiencies, or, if this is impossible, he can alter his project to allow for limitations.

After examining the constraints, the hobbyist can look at the *requirements* for his new system. What does he want the new system to do? Do the requirements for the new system conflict with the constraints? For example, the new system, as originally planned, might require a hardcopy graphics output. However, the hobbyist may not want to spend money for such a device. Here, a budget choice conflicts with a possible requirement.

If requirements and constraints conflict, they should be resolved before proceeding further or they will disrupt the project later on.

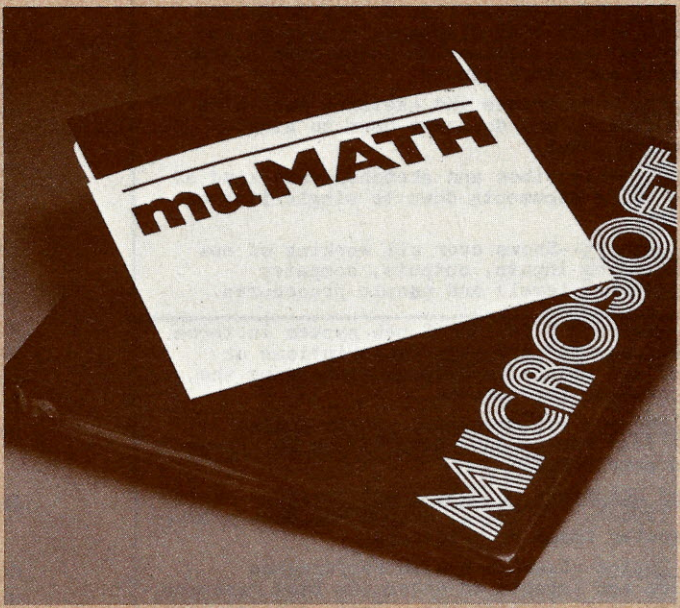
It should be pointed out that the performance description designates output from the system, input to the system, and the external performance of the system, *not* a particular brand of hardware or a particular programming language capability. The performance description should provide specifications which will allow the use of any one of several microcomputers for the project. Normally, a particular computer is chosen later in the project for cost considerations. The results of the analysis phase can be summarized in the performance description shown in Figure 3.

2. *Design Phase* — Here the operations described in the analysis phase are broken into components which will be coded as programs or written up as procedures for the user. The design of a system does not always require new hardware to be purchased. When existing

PROBLEM.

INT(X^N+X*SIN(X^2),X)

SOLUTION.



Surprised? You should be. Because until now, no software could solve anything but the most basic arithmetic problems without a series of complicated steps.

A big disadvantage? If you use a microcomputer for scientific, educational or engineering applications, you bet it is. That's why Microsoft has come up with a solution of its own. muMath.

muMath is a symbolic math package you'll recognize immediately as a major advance in microcomputer software.

muMath lets you efficiently and accurately perform the most complex mathematical operations: Exact, infinite precision rational arithmetic. Unbound variables. Complex expressions (even equations may be included). Exact solution of algebraic equations. Plus logarithmic, exponential and trigonometric simplifications and transformations.

That's right. It does in an instant what took you years to learn at school. Rational arithmetic. Algebra. Trigonometry. Transcendental functions. Symbolic differentiation (including ordinary and partial derivatives.) Symbolic integration of indefinite and definite integrals. Matrix arithmetic and algebra.

Trigonometric simplification? But of course. Just type: ?SIN (2*Y)*(4*COS(X)^3 −COS(3*X)+SIN(Y)*COS (X+Y+#P1)−COS(X−Y)); Then instantly muMath returns: @4*SIN(Y)*COS(X)*COS(Y).

Adding fractions? Need you ask? ?1/3+5/6+2/5+3/7; @419/210.

muMath is written in muSIMP, which is included in the muMath package.

muSIMP is an applicative, recursive language, ideal for describing complex mathematical concepts.

Because of its highly interactive nature and hierarchical structure, muMATH is an excellent math teaching device, from simple arithmetic to calculus.

muMATH is currently available for the CP/M® operating system.

The complete system, including muMATH and muSIMP on disk and documentation is $250. Runs under CP/M.

Just what you need? We thought so. Shoot some questions at us about muMATH. We have all the answers.

Also new from Microsoft: the muLISP interpreter for CP/M. An efficient and reliable LISP system fully capable of supporting serious artificial intelligence efforts. $200.

CP/M is a registered trademark of Digital Research.

MICROSOFT

10800 NE Eighth
Suite 819
Bellevue, WA 98004
206-455-8080
Telex 328945

We set the standard.

Systems, cont'd. . .

hardware cannot support new programs, new hardware must, of course, be acquired.

The newly designed programs and procedures are typically represented in general level flowcharts, descriptions or outlines during this phase. These flowcharts and other design plans form the documentation produced as part of the Design Plan shown in Figure 3.

Up to this point, hardware is not required for the design of the project. The first two phases of a Systems Analysis involve planning and design. Only when this is done, does hardware become part of the project.

3. *Development Phase* — The Design Plan is used as the basis for the programs written during this phase and for the detailed procedures write-up. Further, if new hardware is required, it is assembled, connected and debugged at this time.

Most of the time required for a project, and a majority of the money spent on a project, is used during the development phase. In commercial projects, it is not unusual to have the development phase take 60% of the funds and time for the project.

The Development Phase finishes with the "System as Built" documentation. This is a description of the completed system, including detailed program flowcharts (if they exist), program listings, and hardware descriptions (see Figure 3).

4. *Operation and Evaluation Phase* This is where the new system is put "on-line." Debugging not previously accomplished during the Development Phase is done now. The new system should be fully operational before it is evaluated.

The evaluation of the new system requires a critical comparison between the Performance Description and the actual operation of the new system. If differences exist, are they desirable? Is another iteration through the Design Phase required to make the system work properly?

These four phases are not always clearly defined. In actual practice, the phases frequently merge together with less obvious boundaries. It is rare, though, that a phase is completely omitted. It is not uncommon, even in commercial projects, to return to one of the earlier phases when it appears that the project will not accomplish its objectives.

Documentation for the Hobbyist

When the new system is working as desired and program listings and example inputs and outputs have been collected, the documentation can be finalized. The "System as Built" documentation describes all the items for the completed system. Catalogs, hardware specification sheets, magazine articles which may have helped in putting the hardware together, sketches and handwritten notes used during the building and debugging of the system should be included.

All of this collected information, as well as the Performance Description and Design Plan, can then be put into a large notebook. Figure 4 shows how all of the documentation can be assembled into a single volume. This volume forms the complete documentation of the project, from its beginnings to the detailed description of the finished system. It is used to plan extensions to the system, to aid in maintaining the new system, to help explain your system to others or just to look impressive sitting on your bookshelf!

As you make changes to your system, notes about these changes can be added to the back of the volume to keep the documentation up to date. Extensive changes may require notes on earlier documentation in order to keep system records current.

Analysis Phase

PERFORMANCE DESCRIPTION--Describes performance of new system in terms of input to system and output from system.

1. General Statement--Tells idea of system in paragraph form, what it will do, how it will perform.
2. Document Flowchart--Shows flow of any documents output from system (CRT displays, printouts and other outputs) and information input into system (data sheets and other inputs).
3. System Output--Describes and sketches each kind of output display and document down to single line or entry level.
4. System Input--Describes and sketches each kind of input including documents down to single line or entry level.
5. System Flowchart--Shows over all working of new system including inputs, outputs, computer programs(general level) and people procedures.

Design Phase

DESIGN PLAN--Describes design of new system in terms of detailed computer programs (descriptions or flowcharts) and people procedures processing the inputs to produce the outputs.

1. System Flowchart--If necessary update System Flowchart from Performance Description to change it or add more detail.
2. Layouts of Documents, Displays and Storage--Sketch detailed layouts for input, output and data storage using computer forms or grid paper.
3. Hardware Choice--Examine system Performance Description and Layouts to determine best hardware, if hardware is to be acquired. Cost, availability, and User capability are other important factors.
4. Program Flowchart or Description--Shows detailed computer procedures for program(s) required by new system. Computer code will be written from this flowchart or description.
5. User Procedures--Describes any special tasks which system user must perform. User Procedures are frequently omitted from simple system plans.

Develop Phase

SYSTEM AS BUILT--Describes finished system's hardware, programs, input, output and procedures. This forms record of finished system.

1. Hardware configuration--Describes hardware in sketches and words. Information on all units and custom boards should be included. Vendor catalogs, specification sheets, magazine articles, and hand written notes are important. Catalogs, etc. may be put into special section of completed documentation following this section.
2. Examples of Inputs and Outputs--Descriptions of inputs, outputs, photos or sketches of CRT output, should be included.
3. Program Listing--This includes printed program listings of the latest versions of programs.
4. Notes--Any sketches, notes, calculations, etc. made during the building and debugging of system should be put here.

Figure 3.
Details of the documentation produced at the phases of a project.

Mountain Computer can now EXPAND Your Apple II Peripheral Capacity

EXPANSION CHASSIS

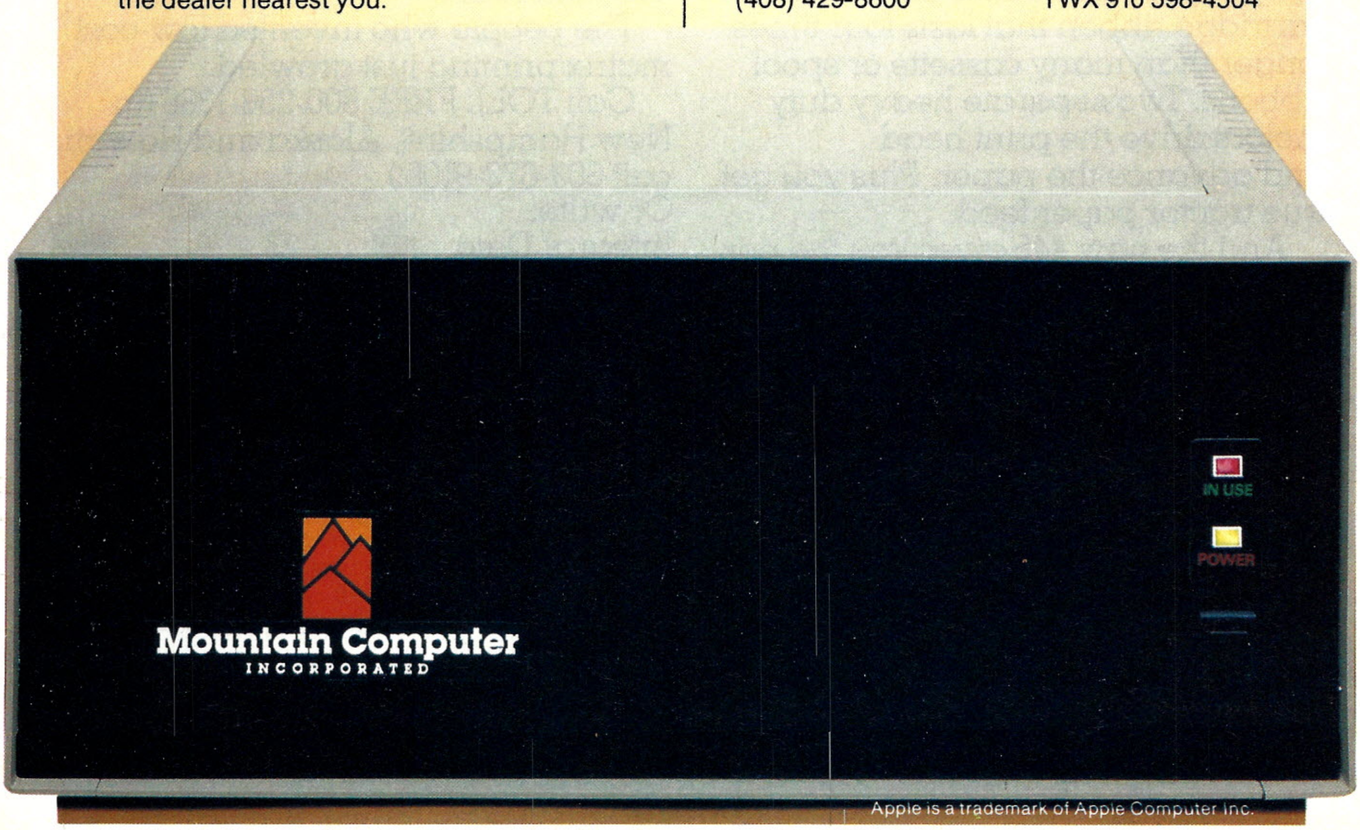
Quality You Expect

Eight more slots for your Apple! Now you can bank-select eight more peripheral slots with immediate or deferred software commands—like having up to 16 peripheral cards "on line"—or use the Select/Deselect switch mounted on the front panel. Expansion Chassis' heavy-duty power supply is primarily for peripherals, without the heavy demand of motherboard support chips required in your Apple. This means much more power is available for peripherals than in your Apple itself! If you've run out of room in your Apple—Expansion Chassis is your answer. Drop by your Apple dealer for a demonstration, or contact Mountain Computer for the location of the dealer nearest you.

Performance You Demand

- Eight mirror image I/O slots of the Apple
- Fully buffered, bi-directional data lines
- Apple II compatible interface card
- Dual selection capability; hardware or software
- Immediate or deferred selection in software mode
- From BASIC, a single POKE command turns the chassis ON or OFF
- Compatible with all software
- Dedicated power supply with approved power transformer

Mountain Computer
INCORPORATED
300 Harvey West Blvd., Santa Cruz, CA 95060
(408) 429-8600 TWX 910 598-4504



Apple is a trademark of Apple Computer Inc.

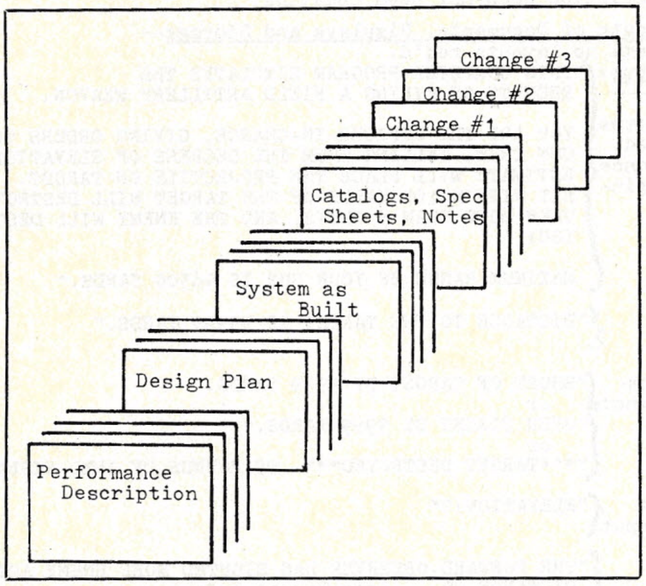


Figure 4.
All of the documentation developed during the four phases of the project, catalogs, specification sheets, handwritten notes and changes to the system after it is operational can be put into a single file or loose leaf notebook as the complete documentation on the system.

A Gunner's Game

PERFORMANCE DESCRIPTION

1. General Statement--Want to write a program for a game to simulate shooting at a target with a cannon. For the first version of the program only the elevation of the gun will vary. If the gunner takes more than 5 shots to hit a target, the gunner and his cannon are destroyed by the enemy. The terrain is assumed to be flat.
2. Document Flowchart--No input data documents will be used for this program. Only CRT output displays will be used in playing the game. No permanent data documents will be generated by this program.
3. System Output--Besides game instructions to the user, the output CRT displays will tell the user if his shots go over the target, are short of the target or hit the target.
4. System Input--The only user input will be the elevation of the gun in degrees.
5. System Flowchart--

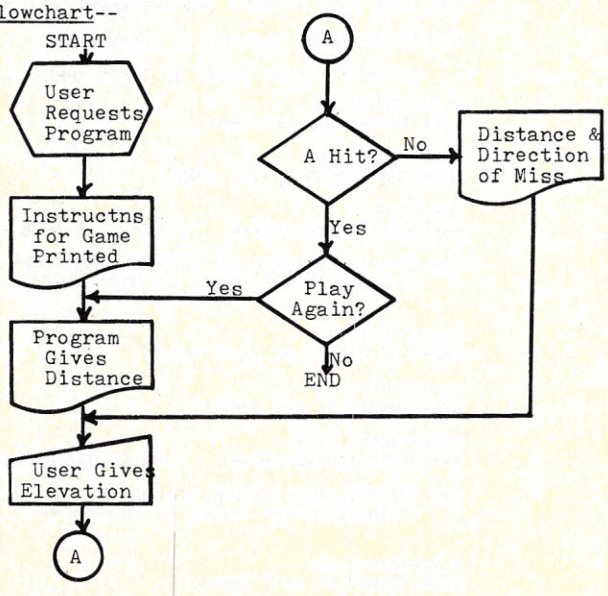
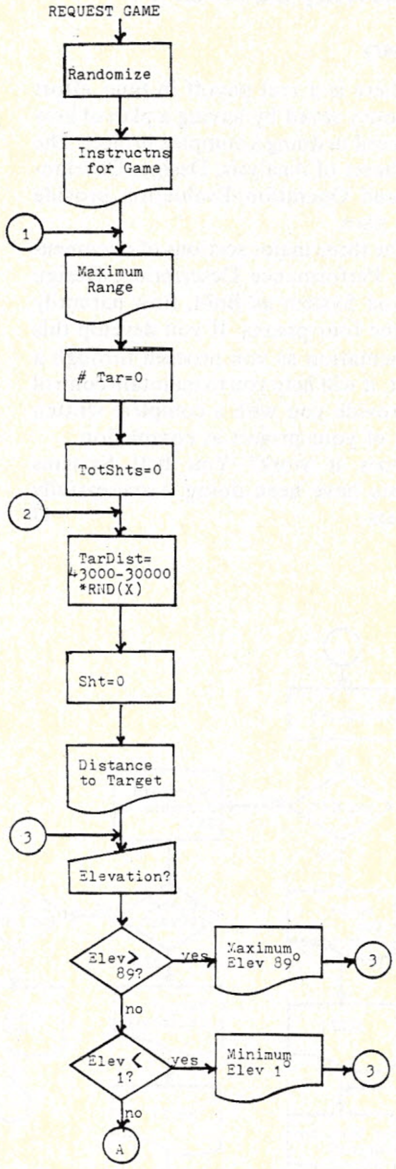


Figure 5.
Example Documentation Produced for a Simple Game Program.

4. Program Flowchart



Example Documentation

Example documentation for a small game program is shown in Figure 5. The illustrated documentation is typed and the flowcharts are carefully drawn to make the pages easier to read. However, when doing your documentation, spend only reasonable time on neatness. Readable, hand written documentation and freehand flowcharts are adequate.

In this short example, the hardware is assumed to be on hand and in working order. No changes have been made in the original system, and no catalogs, specification sheets or other notes have been included. For further understanding, match the example against the general description of the documentation.

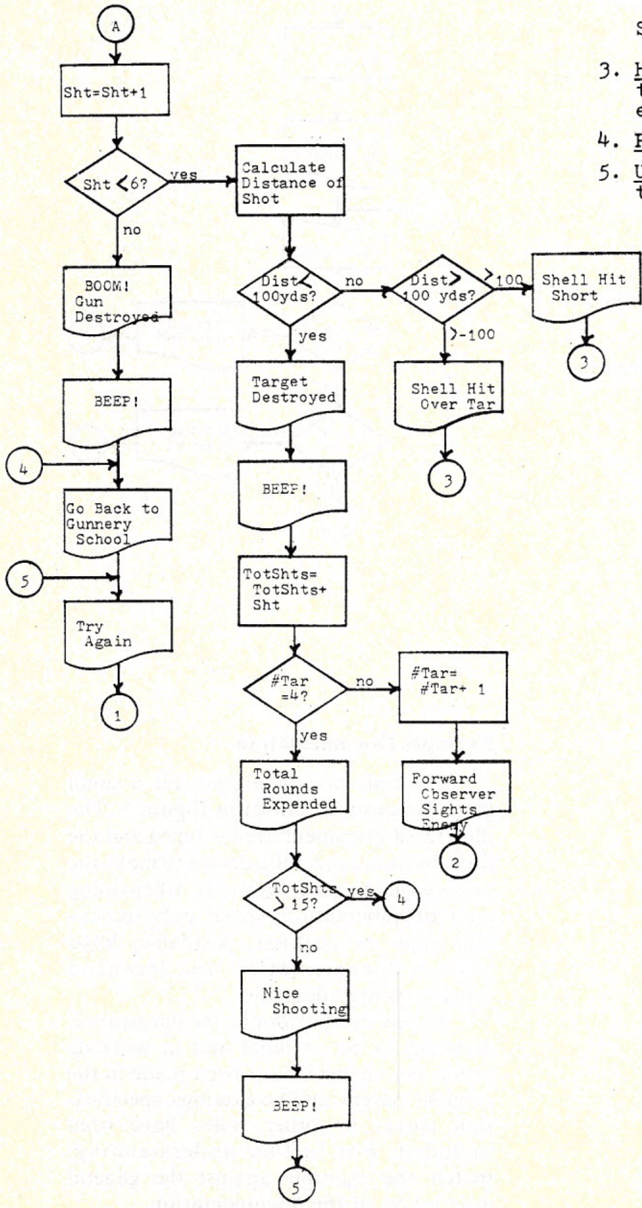
Systems, cont'd. . .

Summary

There is a real payoff in time, effort and money saved by having a plan of how to proceed in doing a complex project. The four phases of Analysis, Design, Development and Operation/Evaluation provide such a plan.

The three major sections of documentation, Performance Description, Design Plan and System as Built, flow naturally from the four phases. If you develop this documentation as you proceed through a project, it will help you to maintain control and provide you with a complete written record of your project at completion.

Does it work? You bet! Systems Analysts have been doing it successfully for years. □



DESIGN PLAN

1. System Flowchart--Flowchart shown in the Performance Description does not need updating.
2. Layouts of Documents, Displays and Storage--
   Input: no layouts req'd.
   Output:

   "THIS COMPUTER PROGRAM SIMULATES THE RESULTS OF FIRING A FIELD ARTILLERY WEAPON.

   Instructns (printed only once at beginning): YOU ARE THE OFFICER-IN-CHARGE, GIVING ORDERS TO THE GUN CREW, TELLING THEM THE DEGREES OF ELEVATION YOU ESTIMATE WILL PLACE THE PROJECTILE ON TARGET. A HIT WITHIN 100 YARDS OF THE TARGET WILL DESTROY IT. TAKE MORE THAN 5 SHOTS, AND THE ENEMY WILL DESTROY YOU!

   MAXIMUM RANGE OF YOUR GUN IS 46500 YARDS."

   Target Info.: "DISTANCE TO THE TARGET IS 99999 YARDS."

   Feedback from shots: "SHORT OF TARGET BY 9999 YARDS." or "OVER TARGET BY 9999 YARDS." or "***TARGET DESTROYED*** 99 ROUNDS OF AMMO EXPENDED."

   Request for Input: "ELEVATION:?"

   Next game: "THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY. DISTANCE TO ...(see Target Info.)

   Storage layout: none req'd. No file records or matices used.
3. Hardware--All of the most common microcomputers will run the program. At least 4K of RAM is required along with either a casette or floppy drive.
4. Program Flowchart or Description--see next page.
5. User Procedures--No special procedures are required for the running of this program.

SYSTEM AS BUILT

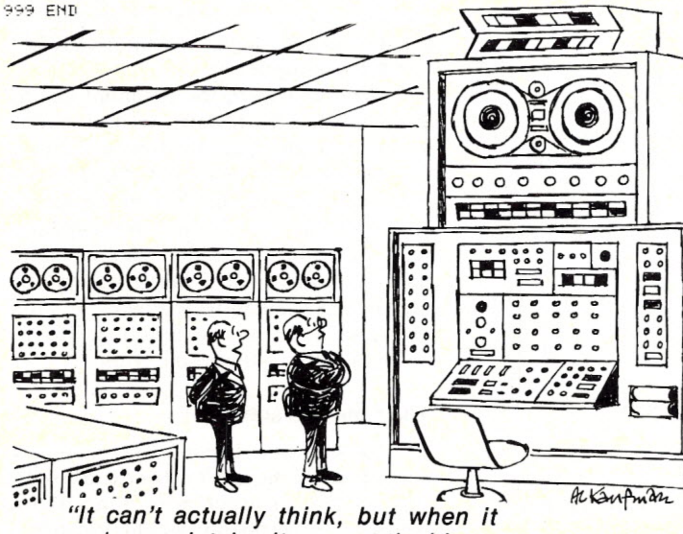
1. Hardware Configuration--The game was implemented on my machine, a Sol 20 with 16K of RAM, cassettes, CRT and printer.
2. Examples of Inputs and Outputs--

```
THIS COMPUTER PROGRAM SIMULATES THE
RESULTS OF FIRING A FIELD ARTILLERY WEAPON.

YOU ARE THE OFFICER-IN-CHARGE, GIVING ORDERS TO THE
GUN CREW, TELLING THEM THE DEGREES OF ELEVATION YOU
ESTIMATE WILL PLACE THE PROJECTILE ON TARGET.  A
HIT WITHIN 100 YARDS OF THE TARGET WILL DESTROY IT.
TAKE MORE THAN 5 SHOTS, AND THE ENEMY WILL DESTROY YOU.

MAXIMUM RANGE OF YOUR GUN IS 46500 YARDS.

     DISTANCE TO THE TARGET IS 23675 YARDS...

ELEVATION:? 20
OVER TARGET BY 6213 YARDS.

ELEVATION:? 15
SHORT OF TARGET BY 426 YARDS.

ELEVATION:? 15.3
***TARGET DESTROYED***     3 ROUNDS OF AMMO EXPENDED

THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY.
     DISTANCE TO THE TARGET IS 17053 YARDS...

ELEVATION:? 10
SHORT OF TARGET BY 1150 YARDS.

ELEVATION:? 11
OVER TARGET BY 365 YARDS.

ELEVATION:? 10.8
***TARGET DESTROYED***     3 ROUNDS OF AMMO EXPENDED

THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY.
     DISTANCE TO THE TARGET IS 31243 YARDS...

ELEVATION:? 25
OVER TARGET BY 4377 YARDS.

ELEVATION:? 20
SHORT OF TARGET BY 1355 YARDS.
```

```
ELEVATION:? 21
SHORT OF TARGET BY 130 YARDS.

ELEVATION:? 21.1
***TARGET DESTROYED***     4 ROUNDS OF AMMO EXPENDED

THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY.
    DISTANCE TO THE TARGET IS 25980 YARDS...


ELEVATION:? 17
***TARGET DESTROYED***     1 ROUNDS OF AMMO EXPENDED
```

3. Program Listing--

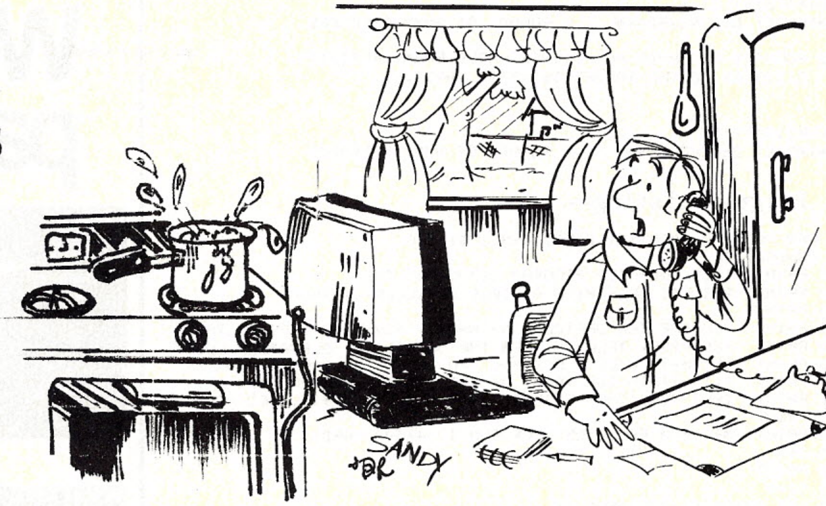
```
10 REM PROGRAM FROM 101 BASIC GAMES BY DEC
90 RANDOMIZE
100 PRINT 'THIS COMPUTER PROGRAM SIMULATES THE'
110 PRINT 'RESULTS OF FIRING A FIELD ARTILLERY WEAPON.
120 PRINT
130 PRINT 'YOU ARE THE OFFICER-IN-CHARGE, GIVING ORDERS TO THE'
140 PRINT 'GUN CREW, TELLING THEM THE DEGREES OF ELEVATION YOU'
150 PRINT 'ESTIMATE WILL PLACE THE PROJECTILE ON TARGET.  A'
160 PRINT 'HIT WITHIN 100 YARDS OF THE TARGET WILL DESTROY IT.'
170 PRINT 'TAKE MORE THAN 5 SHOTS, AND THE ENEMY WILL DESTROY YOU.
175 PRINT
180 PRINT 'MAXIMUM RANGE OF YOUR GUN IS 46500 YARDS.'
185 Z=0
190 PRINT
195 S1=0
200 LET T=43000-30000*RND(X)
210 LET S=0
220 GOTO 370
230 PRINT 'MINIMUM ELEVATION OF GUN IS ONE DEGREE."
240 GOTO 390
250 PRINT 'MAXIMUM ELEVATION OF GUN IS 89 DEGREES.'
260 GOTO390
270 PRINT 'OVER TARGET BY';ABS(E);'YARDS.'
280 GOTO 390
290 PRINT 'SHORT OF TARGET BY';ABS(E);'YARDS.'
300 GOTO 390
310 GOTO 320
320 PRINT '***TARGET DESTROYED***    ';S;'ROUNDS OF AMMO EXPENDED'
322 GOSUB 600
325 S1=S1+S
330 IF Z=4 THEN 490
340 Z=Z+1
345 PRINT
350 PRINT 'THE FORWARD OBSERVER HAS SIGHTED MORE ENEMY ACTIVITY.'
360 GOTO200
370 PRINT '    DISTANCE TO THE TARGET IS';INT(T);'YARDS...'
380 PRINT
390 PRINT
400 PRINT 'ELEVATION:';
410 INPUT B
420 IF B>89 THEN 250
430 IF B<1 THEN 230
440 S=S+1
442 IF S<6 THEN 450
444 PRINT\PRINT 'BOOM!!! YOU HAVE JUST BEEN DESTROYED ';
445 GOSUB 600
446 PRINT 'BY THE ENEMY'\PRINT\PRINT\GOTO 495
450 B2=2*B/57.3\I=46500*SIN(B2)\X=T-I\E=INT(X)
460 IF ABS(E)<100 THEN 310
470 IF E>100 THEN 290
480 IF E<-100 THEN 270
490 PRINT\PRINT\PRINT 'TOTAL ROUNDS EXPENDED WERE';S1
491 IF S1>15 THEN 495\PRINT 'NICE SHOOTING!!'\GOSUB 600\GOTO500
495 PRINT 'BETTER GO BACK TO FORT WILL FOR REFRESHER TRAINING!'
500 PRINT\PRINT 'THANK YOU FOR PLAYING!'
505 PRINT\PRINT 'TRY AGAIN....'\PRINT\GOTO 180
600 FOR N=1 TO 10\PRINT CHR$(7);\NEXT N
610 RETURN
999 END
```



"It can't actually think, but when it makes a mistake, it can put the blame on some other computer."

©Creative Computing

The Bleak Future of Small-Business Computing



Paul F. Doering

Beware of predictions when experts agree. Opinions aside, Titanics do sink and swine flu epidemics fail to materialize. Today's "everybody-knows-it" is that the golden age of small-business computing is nearly upon us. Rubbish. We're on the threshold of unprecedented confusion.

"Small-business computing" is the use of cheap, dedicated computers to automate the recordkeeping of companies with low dollar volume and few employees. Numbers for *low* and *few* vary with the user, but let's include Bertie's Button Boutique and exclude Chrysler.

The Problem

Writers and lecturers tell us that the availability of modestly priced systems makes it practical for small companies to rely on computers for accounting, freeing people to concentrate on managing. Sounds neat, eh?

The problem is that the accounting *is* the managing. Once you've got all those figures for factors like inventory, receivables, payables, and profit projections, management consists of doing what the bottom line demands. Oh yes, managers also hire help, select product lines, and decorate showrooms; but it's ignorance of that bottom line that leads to failure. That golden era we're approaching is actually one of automated ignorance.

The Timebomb

You see, it's the word "cheap" that represents the timebomb. The $5,000 disk-based computers are really good values. A competent commercial programmer (humor me and accept the abbreviation CCP) can convert them into systems from the experts' dreams. Most systems will be programmed instead by kitchen-table amateurs (KTA's).

The CCP recognizes that systems must be maintained. He prices his product to allow revisions whenever the governments change their reporting rules. He incorporates the true cost of money in his projections. He includes automatic file backup. He keeps an audit trail. He alters a record only after confirming the validity of the new item. He offers editing for data entry. He tailors the prompting messages to the operator's level. He generates paper records as a basis for manual accounting during unscheduled downtime. He documents exhaustively. These features and others all add to the initial cost of the CCP's product.

Opinions aside, Titanics do sink and swine flu epidemics fail to materialize.

The KTA overlooks some of these niceties. It's enough for him that his inventory program calculate the number of marshmallows on hand. His program is cheap. Tick, tick, tick.

The Mad Bomber

There is no practical limit to the supply of bad programs, because our hobby is spawning a large cottage industry. I know a few programmers who genuinely promote computing. They conscientiously develop tight, fast, reliable packages which meet the user's expectations. These careful programmers are uncommon.

Many programs sold today are coded quickly, loosely, and carelessly. There is only superficial testing. No one confirms that every loop has been tried, every meaningful data mix injected. No one traces the calculations for possible loss of numeric significance. If it works once, it must be okay. Anyway, selling a few copies will yield enough money to buy a printer to check the coding.

Much of what gets published in computerist magazines promotes this attitude. It is tempting, isn't it? Form a company with a fancy name, sell a program, write off your equipment as a business expense, get rich. Everyone's doing it. Then you find out that the bottom has fallen out of the games market, so you turn to the next touted mother lode: small business. You borrow a book on accounting, dash off an implementation of your untested understanding, and rush to advertise. We'd all be horrified if anyone did that with a drug or a vehicle, but it's far more common in programming than you believe. Is injuring the means of someone's livelihood less reprehensible than injuring his body?

The Odds

Do you doubt that there are more bad programs than good ones? Look at what happened to radio discipline when CB became a toy. That's what is going on today in programming. Many suppliers teach programming. Very few stress responsibility and the need for self-discipline. With a stream of questionable programs reaching the marketplace, the chances of choosing a dangerous package are disturbingly high. The ratio of KTA's to CCP's is topheavy.

Who will warn the potential victim? Probably no one. If Bertie of button fame knows enough about the financial end of his business, he may cut through the mystique and be hardnosed about his specifications. He may ask about updates, fixes, revisions, warranty, and a maintenance contract. But if Bertie knows only buttons, he's apt to believe what he's told about small-business computing. Mostly, he's not hearing the whole truth.

The Consequences

Let's be fair. The manager intelligent enough to see the merit in the extra cost for the CCP's product will benefit and prosper. He probably would prosper in any case; the computer will merely enhance the prospects for success.

Paul F. Doering, 56 Elmore Road, Rochester, NY 14618.

Now, you may object that a business run by someone ignorant of sound practices will fail anyway. Perhaps, but the computer is being promoted as insurance against such ignorance. Read the ads. The CCP and the KTA are making similar claims. Both promise to automate a crucial aspect of the enterprise. Neither lists the features omitted. Neither specifies what

The CCP and the KTA are making similar claims. Both promise to automate a crucial aspect of the enterprise.

supplementary manual effort is required or what minimum knowledge the user must possess. The result of a wrong choice is that the wasted investment in the KTA's system will hasten the failure of the weak business. Yet where do we read that the coming of small-business computing will boost the failure rate?

I wish I could tell you that the solution lies in buying software from the major hardware vendors. Sadly, some of them apparently employ their own KTA's. As an example, a mailing list program from a leading personal computer manufacturer takes 30 hours to sort a list of 150 names and addresses. I know a CCP whose version in the same Basic takes only 55 seconds.

How many small businesses could tolerate the day-long loss of inventory-, ordering-, and accounting-capabilities just to update a mailing list? Guess what? The major vendor's software specifications don't mention this problem. Furthermore, if you trigger this sort, you'll have no clue about how the routine is progressing or when you might expect to get your system back.

Slothful performance has one benign aspect: the user can detect it. It may disquality that particular system, but at least it's not fatal.

Sloppy data-handling, though, is a disaster. I've seen a program that can't abide a key's being struck during a file update. The consequence of such an innocent act is the substitution of that key's value for the one then being moved. There's no warning; the file is contaminated and accepted. That is a program's ultimate transgression.

If the manager is lucky, he will notice that an item makes no sense. Whether he will then be able to correct it is still an open question. Should the substitution go undetected, the timebomb has been armed and started. The computer may later order a wrong item, misaddress a shipment, report erroneous FICA data, overpay a supplier, underbill a customer, or erase the disk. Even a capable manager could lose a business with that kind of help.

Before you laugh, how does your system handle that situation? How do you know?

The Remedies

What recourse is open to the person whose business dies from contaminated data? Perhaps he can sue the KTA, but with what real hope of getting remuneration? Many KTA's are at least clever enough to include disclaimers about fitness, merchantability, and nonresponsibility for consequential damages. Realistically, there are no after-the-fact remedies.

Are there any safeguards for the prudent buyer? I usually offer this personal advice:

It's often wiser to buy your cheap computer from someone scaling his computers down, rather than up. I think your chances of success are better if you pay more to IBM than if you pay less to a fledgling concern known for TV games or calculators.

Never discontinue your manual practices until your computer has demonstrated that it can do as well. Run the manual and the automated operations in parallel for a while.

Ditch the computer if it doesn't perform from the beginning. Your initial dissatisfaction arises from only the bugs you've been able to detect. If obvious things are going wrong, don't risk assassination by the hidden bugs.

Never buy a computer package without first interviewing experienced customers. Talk to the computer operator as well as the company owner. Don't rely on hearsay testimonials. Remember that you're in an adversary relationship with the salesman. It's like buying a car.

Don't expect the system to compensate for your own ignorance. How will you know if it's working?

Remember that you're in an adversary relationship with the salesman. It's like buying a car.

Find out whether the vendor is using this system in his own operations. It's no guarantee, but it's a devastating question to spring on the salesman.

The Prospect

Too few managers will demand rigorous proof of performance before committing their company's well-being to a computer system. We who relish the beneficial impact of computers on mankind will sustain bruises during the coming wave of disillusionment, as small-business computing too often betrays its promise.

You can lessen the blow by speaking out on the need for responsibility in programming. □

Stan Gets Re-Randomized

David Katz

This is the Stan Story to end them all. Yup that's a promise — I can't Stan them either. But it's a fun way to learn. — Ed.

"Say, Stan, come on into the den for a while. I want to show you something," Stan's father said as they were getting up from the dinner table one evening. "It's about that program we developed back in February to study the probabilities from rolling dice."

"That sounds interesting. Can I come and listen too?" his mother asked.

"Sure, Mom. Then, after we're done, I'll help you with the dishes."

"!!!"

As they were walking into the den, his father started to explain. "You see, son, I was looking at the probabilities you produced with your program. While they were correct, something bothered me about them. Why would the total of the wins and the losses on 180 rolls of the dice also be 180? Some of the rolls are neither winners nor losers in a real game. For example, if I roll a 4 and then a 9, that 9 doesn't win or lose. Since I'm looking to roll another 4 to win or a 7 (I hope not) in which case I lose, the 9 is just a wasted roll. Actually, anything other than a 4 or a 7 is wasted."

They were in the den now, and while Stan turned on the computer and got the program ready, his Dad continued, "In a real craps game, you would expect some of the rolls to be wasted like that. Therefore, you would expect less than 180 for the total of winners plus losers. Don't load that copy of the program. I have another copy I've made some changes to, and I want to show you what happens."

Stan stepped aside to let his father load his own copy of the program.

"Tell me what your program used to do while Dad is loading his," Stan's mother said.

David Katz, 1526 Empress Ave., Saskatoon, Sask., Canada. S7K 3G3.

"Well, Pop was trying to write a program that would simulate the rolling of dice. And he didn't think that the patterns he got from the random number generator were very good. So I decided to load a one-dimensional array of 36 elements with the values from 2 to 12 in the same proportions that would show up from the dice. There was one 2, two 3's, and so on. Then, in order to generate a roll of the dice, I would generate a random number between 1 and 36 and use that to choose one of the possible rolls. I remove that number from the list and next time generate a number between 1 and 35. By doing that, the program produces rolls for the dice which exactly match the probabilities one would predict."

"Do you mean that you do it like dealing a deck of cards where you take the value out and don't replace it?"

"Yeah. That is exactly how I described it to Dad, only he said that one set of rolls wasn't enough, so I wrote another program to do exactly the same thing — only it had five times as many values. The list was 180 numbers long."

"But the numbers you generated weren't *stochastically independent!*"

"Huh! What is that?"

"Well . . . Let me try to explain it this way. Suppose we take the example in which you were using only 36 rolls. Now, the combination for boxcars (two 6's) only has a probability of 1 in 36 of occurring, so it is only given once in your list. Right?"

"Yup."

"So, if the first number I roll is 12, then it gets removed from the list and I can't roll it again, at least for the next 35 rolls. Right?"

". . . uh, I think I can see what you are getting at. There is nothing in real dice that says you can't roll another 12 right away. But I took care of that when I had the list with 180 values. With that, you could roll another 12 on your next roll and nothing would be wrong with that."

"True, but nothing in the dice themselves says that I couldn't roll six 12's in a row either."

"But my program would only allow you to roll five 12's at the most. OH-OH!"

"Right. Dice are stochastically independent. That means that whatever happens on any roll is not influenced by any previous roll and will not affect what you get on any other roll."

"But the numbers that Dad was getting from his program didn't look very much like the kind of results you would get from a pair of dice either."

"Do you have a print out of those numbers around?"

"Over there — on top of that pile. I ran off a copy of his results before I started playing with the program."

By now Stan's father had his program listed and was ready to run it, so both Stan and Mom waited for him to explain what he had written.

"Take a look at what I did here before I run the program. I changed your program a bit, so that instead of generating a random number in lines 530 to 540, I take the counter that you use in line 120 to decide if another random number is needed and subtract it from 181. Then I use that instead of the random number. You can see that the rest of the program hasn't been changed at all."

"Okay."

"Now then, let's run this new version." The whole family stood bathing in the glow of the display for a few minutes while the program ran, and pretty soon . . .

"Heh! Those are exactly the same numbers I got with my random number generator. But you aren't using random numbers."

It was about this time that Stan's mother picked up the hand calculator and wandered over to the small desk in the corner of the room.

"That's right, Son. Your program didn't take into account successive rolls after a point was made, so you were counting the results from possible out-

Stan, cont'd . . .

comes for the first rolls only. That also explains why the total of winners plus losers was exactly 180. In lines 304 to 346 you divided rolls for points according to their probabilities without actually rolling the dice any more. You only used the ratio of wins to losses for a given point. Your program really shows what happens in 180 TURNS!!!"

"But if that's the case, I could have done it without the computer. I know that on the first roll 8 out of 36 win with a 7 or 11, and 4 out of 36 lose with 2, 3 or 12 and I had the correct probabilities for the chances of a win for any of the other possible rolls. I could have just gone ahead and calculated the result without the computer."

"Right again. Since the 4 will show up 3/36 of the times on first roll and the chances of winning after you have a point of 4 are 3/9 then you can add 3/9 times 3/36 or 1/36 to the chances of winning and 6/9 times 3/36 or 2/36 to the chances of losing. Then all you have to do is complete that for all of the other points that take more than one roll."

At just that moment, with the timing that can be found only in Fairie Tales and computer magazine scenarios, Stan's mom returned.

"As far as I can tell, this all started when Dad sent you off on a wild goose chase, Stan. I just finished trying a Chi-Square analysis of the original numbers your father had, and they are within the kind of distribution one would expect from real dice in 100 rolls."

"What is a Chi Squared watchumacallit?"

"Let me explain. Start by imagining a pair of dice that always gave you the exact probabilities for each of the numbers for every set of 36 rolls. Would you say that was a fair pair of dice?"

"No," said Stan and his father at the same time. Stan continued, "That sounds like the random number generator we just had — which wasn't too good."

"And if you had a set of dice that gave six 2's, five 3's, four 4's, three 5's, two 6's, a 7 and an 8 and then two 9's, three 10's, four 11's and six 12's?"

"Those sound like loaded dice," replied Stan's father.

"Good. What we would like to see is a set of dice that produces something like the expected probabilities, but not exactly those proportions. That is the way a set of dice should behave. The question is how to tell if the rolls that we get are close enough to real dice to be acceptable, but not so close that the dice must be rigged? That's where statistics comes in. We can test some rolls of the dice to see how 'real' the results are. If the dice were rolled 100 times, as in the sample test that Dad had, one would expect to see the value 2 showing up about 1/36 times 100 or 2.78 times. That's according to probability. In the test, there were two occurrences of the value 2. Now, in Chi Square, we call the number that we got in the test the *observed quantity*. We calculate the value of the observed minus the expected quantities and square that. This result is then divided by the expected quantity. We do that for each of the possible combinations and add them all together."

"That's all very interesting, but what good is it?"

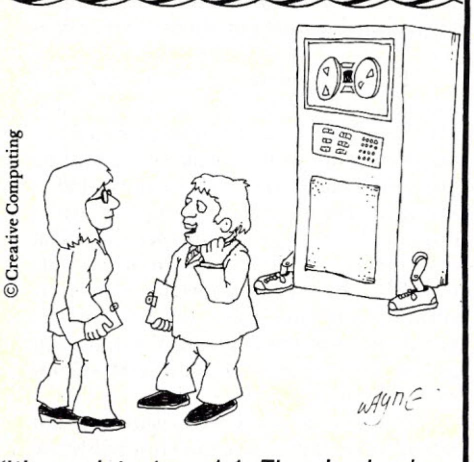
"Well, if the result we get is 0, then all of the observed values are exactly the same as the expected values, and we already know that is unacceptable. On the other hand, if the value is really large, then the observed values are way out of line and that is no good either. What we want is some value that is in the middle. If you look in this book (Knuth, D.E., 1969; *Seminumerical Algorithms;* Vol. 2 of the seven part *The Art of Computer Programming;* Addison-Wesley Publishing Company, Reading, Mass.) where the author talks about random number generators and testing them, you will find a table that tells you what range of values are acceptable for a dice simulation (pages 34-39). If the value is between 3.94 and 18.31 then the simulator behaves in about the same way as a pair of real dice would. In the case of the Dad's test data, the value was 14.68. That is extreme enough to suggest trying the statistic on a few more examples, but it looks pretty good."

"Can I use this Chi Square stuff to test other kinds of random number sequences?"

"Yes. You might have to use a different line in the table if you are producing a smaller or larger number of different values, but the general idea is the same."

"Well, I guess you and I owe Mom one this time," Stan's father said. "Seems we should have asked for her help earlier."

"Well," Mom said, "I wanted to try a few more tests on the random numbers, and there's still the matter of some unwashed dishes. So . . ." □



"It's our latest model. There's absolutely no problem it can't solve or run away from."

Interactive Systems And The Design Of Virtuality

The Design of Interactive Systems, Tomorrow's Crucial Art Form, Rests On New Philosophical Principles

Ted Nelson

What follows expresses the personal views of the editor.

It should be plain by now to anyone that the future of mankind is at the computer screen. Already hundreds of thousands do their work at screens, working with text or diagrams or maps.

This is only the beginning, and I think it is obvious that virtually every form of non-physical work, and many forms of play and leisure study, will soon migrate to the interactive computer display screen, as suitable systems are designed.

Yet most interactive systems are lousy; and almost every new interactive system I see leaves me aghast.

In this article I will try to set forth very briefly, with examples, what I conceive to be the correct principles of design for interactive systems. There is much more to be said, at both concrete and theoretical levels, but here for now is the brief overview that is so sorely needed.

Interactive screens, as everyone should know, can present anything — text, pictures, maps, the latest information on whatever you need to keep track of.

Real-world control from a screen already occurs in some places. Systems are installed or available for factories and utility companies — and military forces — that allow the user to modify the world at the touch of a picture. By changing a picture, the user can cause the real world to change accordingly — if the system is so set up. Change a valve in a diagram on the screen, and the real valve itself, half a mile away in the refinery, opens or closes.

The potential flexibility of such systems is enormous. Any pointing tools can be used to work in this fashion, if suitably linked into the controlling computer and its program. (You could open that valve with a keyboard, a joystick, a tablet, a trackball, a mouse — or any other mechanical hookup you might prefer.)

Both laymen and computer people mistakenly believe that the design of interactive systems is somehow a "technical" matter. While there are many technical aspects to such design, I believe the conceptual and even artistic problems of such design far outweigh in importance the mere technical details. The designing itself is art, not science at all. (Sometimes, of course, there is a question of whether a thing can actually be done, or how, in which case technical questions loom large.)

PITFALLS

There are many design pitfalls in attempting to build interactive systems. Let us run by these quickly.

1. Cheap AI Systems

One approach that was popular five or ten years ago was the "artificial intelligence" approach, based loosely on the idea that a user would type in English-like sentences, and the program would pretend to be alive, friendly and understanding in responding to what you thought you asked for.

But the typing of input strings is tedious and generally a waste of time. Moreover, the program's masquerading as an intelligent entity is usually misleading and annoying, both for the time wasted in trying to guess what the program *really* does, and for its gratuitous pseudo-social invasion of contemplative privacy. Fortunately, this type of system is proportionally on the wane (except in the personal computing field, where it lives on as the "adventure," or Guess-My-Commands, game).

(Note that these criticisms are not intended to apply to artificial intelligence as a long-term goal, but only to artificial intelligence as a local pretense of contemporary software.)

2. Command Languages

Another common approach is to create a command language for what you want to do — like, say, the "text editors" of old. Each action must be called forth by typing in an input string in code.

But user languages are hard for beginners to learn, and have the "computerish" feel that is so repellent to non-computer people.

Command-language systems are also clumsy and obtuse compared to more highly responsive design. (As one widespread example, more highly interactive forms of word processing have appeared, supplanting the old text editor programs that used to require (for instance) explicit insertion commands.)

Command languages are also dangerous. By permitting many simultaneous options (and variations, and modifications of further variations), command languages make it possible for things to go terribly wrong in a very short time — terribly, terribly wrong; irrecoverably wrong. While language facilities must of course be available to programmers, an environment in which a user thinks *about something other than the computer* should not be tangled by the complications of a command language that forces his attention where it does not belong.

3. Menus and Afterthought Interfaces

The last pitfall is what I call the menu trap. Now, menus are all right, and better than the approaches I have mentioned so far, but for high responsiveness and performance values, I believe the kinds of interactive systems we will describe below are greatly preferable.

The so-called "user-friendly interface" is a variation of the menu trap. The very phrase suggests that the user interface goes on after the system is actually built, like paint. We will speak more about this later.

The Ten-Minute Rule

An interactive computer system for most purposes should be learnable in under ten minutes. This criterion shocks many computer people. It has certain vaguenesses. But as a striking statement — a battle cry — mandating high-power interactive design, it is the most concise way I know of saying how easy things should be. They understand the ten-minute rule in the arcades. They understand the general idea of virtuality design in the arcades. The people who don't understand are the "computer professionals" — who sometimes do great damage.

VIRTUALITY

The central concern of interactive system design is what I call a system's *virtuality*. This is intended as a quite general term, extending into all fields where mind, effects and illusion are proper issues.

By the virtuality of a thing I mean the *seeming* of it, as distinct from its more concrete "reality," which may not be important.*

An interactive computer system is a series of presentations intended to affect the mind in a certain way, just like a movie. This is not a casual analogy; this is the central issue.

I use the term "virtual" in its traditional sense, an opposite of "real." The *reality* of a movie includes how the scenery was painted and where the actors were repositioned between shots, but who cares? The *virtuality* of the movie is *what seems to be* in it. The *reality* of an interactive system includes its data structure and what language it's programmed in — but again, who cares? The important concern is, what does it *seem to be?*

A "virtuality," then, is a *structure of seeming* — the conceptual structure and feel of what is created. What conceptual environment are you in? It is this environment, and its response qualities and feel, that master — not the irrelevant "reality" of implementation details. And to create this seeming, as an integrated whole, is the true task of designing and implementing the virtuality. This is as true for a movie as for a word processor.

The virtuality of an interactive system is composed of its conceptual structure and its *feel*. A system should have both a good conceptual structure and the right feel.

The truly interactive system, as in the arcades, needs no carriage returns; each user action creates an instant response — and may not echo what you typed.

(It is amusing to note that the firm most associated with "the computer" in the public mind does not manufacture computers which can be programmed in this way. Yet I believe this is how computers should in general be programmed.)

Here too we see an exact similarity of the interactive system to the motion picture. It is the overall impression, not the component parts or the particular tricks of presentation, that count. In a movie it doesn't matter what kind of camera or form of scenery was used to make a given shot; what matters is the contribution that the scenery makes to the shot (and its feel), and the contribution that the shot makes to the film (and its feel).

The interactive system, I think, may best be thought of as a new kind of movie; but a movie that you control, and a movie that is about something you wish to affect. And it is the imagining of this interactive movie that is the important design task. Never mind the nuts and bolts — what's the dream?

The following brief discussions are intended to highlight some interesting systems and their special features, considered as virtualities. In the final part of the article I will endeavor to tie together some general principles of virtuality design.

A more detailed analysis would get down to exact controls and the way they are merged with presentations, showing the infuriating intricacies of ramification that must be dealt with. I could spend a page comparing the cursor behavior when two different word processors Delete, and much more on the nature of conceptual structure. But not in this overview.

DATALAND

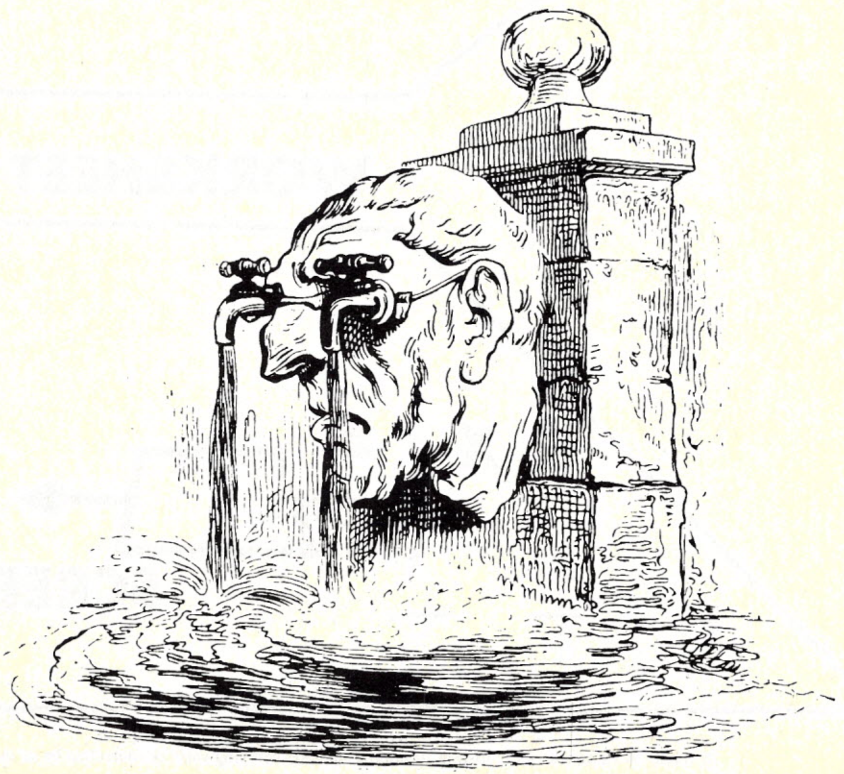
One of the most striking and easy-to-use computer systems in the world is the system called "Dataland," created by Nicholas Negroponte and his associates at the Architecture Machine Group at MIT.

The Dataland user sits at a large screen with various controls. The main controls are essentially joysticks that allow you to pan and zoom.

On the screen you see an unusual collection of pictures, some quite small. You may zoom in on anything (panning the screen to select what you want magnified), and the original pictures or symbols will be enlarged, augmented by more detail, or replaced by other pictures.

In principle this can go on indefinitely. It is like being in satellite orbit with a huge zoom lens, able to look at an overview of Los Angeles, then magnifying it until you can read the fine print on a newspaper on the sidewalk.

Because Negroponte is a great showman, he has dolled the system up with a variety of graphic features, synthesized on the screen by computer, such as ticking clocks.



*The closest other term I can find is "mental environment." My students have urged me to retain the term "virtuality," even though it causes confusion among users of so-called "virtual systems," meaning real systems configured with virtual huge memory.

By storing detailed information on any topic in the Dataland at specific levels of magnification, it is possible to put huge arrays of data where the user can essentially pick out what he wants graphically and dive into it. A stunning, sweeping concept, rather obvious if you think about it; yet it makes many of the systems that have been developed for database query look foolish. You can do it all visually with Dataland, at least with hierarchical data. In under ten minutes' training, you are able to search through great hierarchies of stored information, pictorially.

Although the system in its present form requires millions of dollars worth of equipment, I am sure that within a few years we will have access to comparable systems on machines as small as the Apple or Atari.

Enthusiasts have supposed that there is something fundamental and magical about Dataland's two-dimensional array. From the standpoint of virtuality theory, however, the 2D array is only one kind of clear and simple world. This system's power simply shows the power of two-dimensional organization in our thought, especially because of our experience with paper and other 2D viewing situations. But there are many other powerful organizing ideas. While clarity and simplicity (and good examples like Dataland) are desirable, many other screenworlds will also be exciting and useful.

VISICALC

One of the most important new software products is Visicalc, originally designed by Dan Bricklin and now marketed by Personal Software, Inc.

Visicalc (see review in last August's issue of *Creative Computing)* effectively creates a vast dynamic worksheet for bookkeeping, accounting and financial planning.

Now, many people suppose that accounting is an exact science. Many accountants, however, feel differently. A great deal of their time is spent reworking columns of figures and deciding which numbers belong where. An accountant may spend hours creating lists of expenditures, adding them up — then removing items, putting other items in, and adding the column up again in its new form.

This is all perfectly legitimate. Just *why* accountants do this all the time is outside the scope of this article, but they do.

Visicalc consists of *programmable columns.* You may create, for instance, a column of figures, and tell it to keep a sum at the bottom that is always correct; every time you insert or remove a new item, the sum is recalculated.

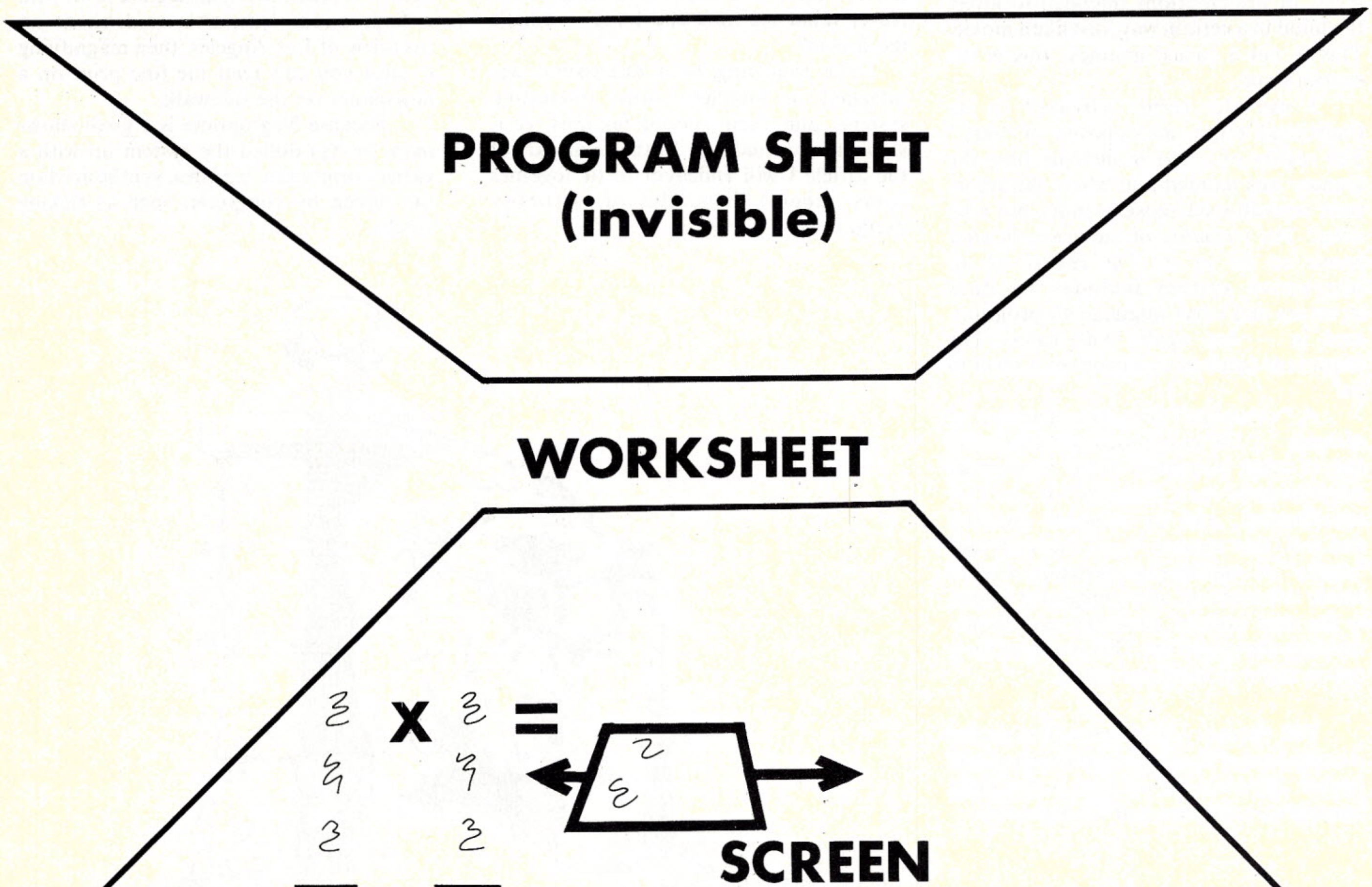
But Visicalc goes much further. You can tell this programmable worksheet, for instance, to take 20% of each figure in Column A and put it in a corresponding position of Column B.

The programming is by example, and somewhat hard to learn, but Visicalc essentially allows the creation of an enormous array of columns and terms, with fairly complex relations among them, all instantly up-to-date every time you change a figure.

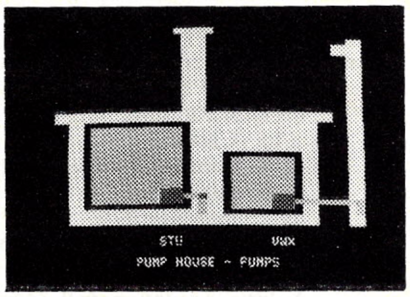
Those who work a lot with figures consider it breathtaking, and there are accountants who consider it reason enough to get a computer.

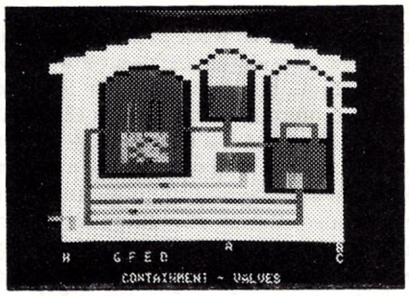
Visicalc might be visualized in any number of ways. I like to think of it as two sheets — the columns of figures, which can be seen by the user, and the overall program sheet, which cannot.

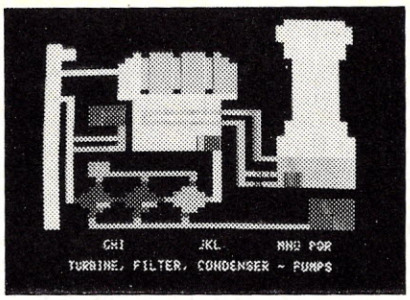
Thus the *world* of Visicalc consists of

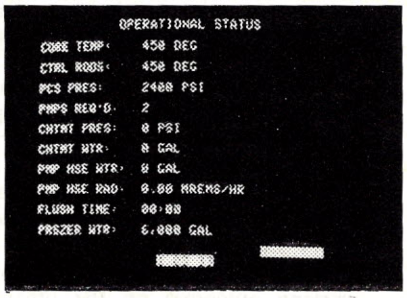


A sheet of columns, with instant summing-up and carrying-over, is controlled by an invisible program on an unreachable sheet.



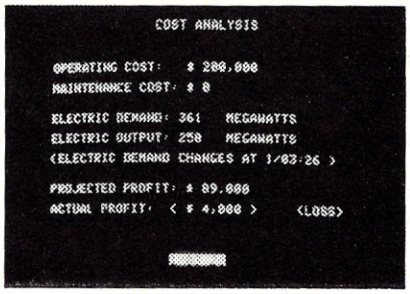






THREE MILE ISLAND

by Richard Orban



Put yourself in control of a nuclear reactor!

"The aids you have in operating the plant are excellent. They include four graphical views of portions of the plant, an instrument panel, a financial summary and an equipment status and failure log.

"As a simulation Three Mile Island is excellent. As a game it is fascinating to me."

Victor Fricke, Nuclear engineer in the March 1980 issue of Creative Computing.

"This program will be a classic.

"No school should be without Three Mile Island. If parents are buying a computer for the personal development and education of their children they must buy this program."

Robert Elliot Purser, publisher of Purser's Magazine, A Software Directory.

For the Apple II and Apple II Plus (48K)

$39.95

Apple is a trademark of Apple Computer Corp.

MUSE SOFTWARE™

330 N. CHARLES STREET
BALTIMORE, MD 21201
(301) 659-7212

CIRCLE 226 ON READER SERVICE CARD

Call or write for a FREE catalog or the name of your nearest MUSE dealer

these two sheets; but the *views* allowed of this show only the lower one. (This is something like the division between Heaven and Earth in some religions, the former supposedly being visible and the latter not; but I would hesitate to conclude anything about the religious views of Visicalc's designers from this.)

Considered as a virtuality, then, Visicalc has a serious blind spot: although you create and use the program sheet, you are not allowed to see it.

Pointing up such omissions is a possible benefit of thinking in terms of virtuality. By considering the whole shown world, its views and feel and control structures, we can search out omissions and asymmetries and suggest better structuring. (That this is possible for something as good as Visicalc shows that this approach may be of very general benefit in the future.)

PARKED CARS

An unusual example of an elegant and spare virtuality is a work entitled "Parked Cars" by the Argentinian artist Laszlo Snead. It has not previously been described in print. While the work is still only in the planning stages for the Apple, it can be considered as an interesting virtuality whether or not it is ever completed.

Some might call "Parked Cars" an Adventure game; the artist himself prefers to refer to it as a "work of art." Indeed, it might be considered an X-rated work of art, since it may include both sex and violence, but only if the user so chooses.

"Parked Cars" is to be in the form of a comic strip, showing one panel at a time in Apple lo-res graphics. (This mode on the Apple also allows an independent rectangular panel of scrolling text, which may serve as talk balloon or caption). Snead intends the overlays for the whole of "Parked Cars" to just fill one side of a diskette, so there is to be much doubling-up of text and graphics, in ways the artist hopes will be inspired.

The action of "Parked Cars" takes place at a scenic overlook on a parkway at night. Three cars drive in, in random order. They are inhabited, respectively, by persons that Snead refers to as "The Hot Couple," "The Nervous Couple" and "The Guy With a Knife."

Now, a number of interactive stories already exist for reading from computers. This one may be the first to have extensive graphics. What is especially interesting from the virtuality standpoint, however, is Snead's selection of the user controls, and the way they relate to the world we are watching.

The user may simply rove through the three-dimensional scene as a disembodied spirit, spying on the different characters. But they won't do much; they stay in loops. The user may, however, *enter the psyche* of any character, and read that character's impulses as they pass through his or her mind. The full set of commands is:

ROVE AS SPIRIT
ENTER PSYCHE/LEAVE PSYCHE
THINK (brings forth a thought to read)
ACT ON THOUGHT
GO TOWARD/AWAY FROM OTHER CHARACTER

A little cogitation will suggest the vivid potential of this small set of commands, playing through a suitable scenario — which Sr.Snead is well advised to supply.

What is less obvious is that this interesting control structure is especially suited to the somewhat raw material that Snead has chosen. While the same control structure could be applied to a more urbane dramatic setting — say, a detective story or political melodrama — the potential tedium on the one hand or wildly varying outcomes on the other would create great difficulties. The choice of a small cast in a highly-charged small setting would seem to be ideal balance for working through the artistic premises of this virtuality.

WORD PROCESSORS

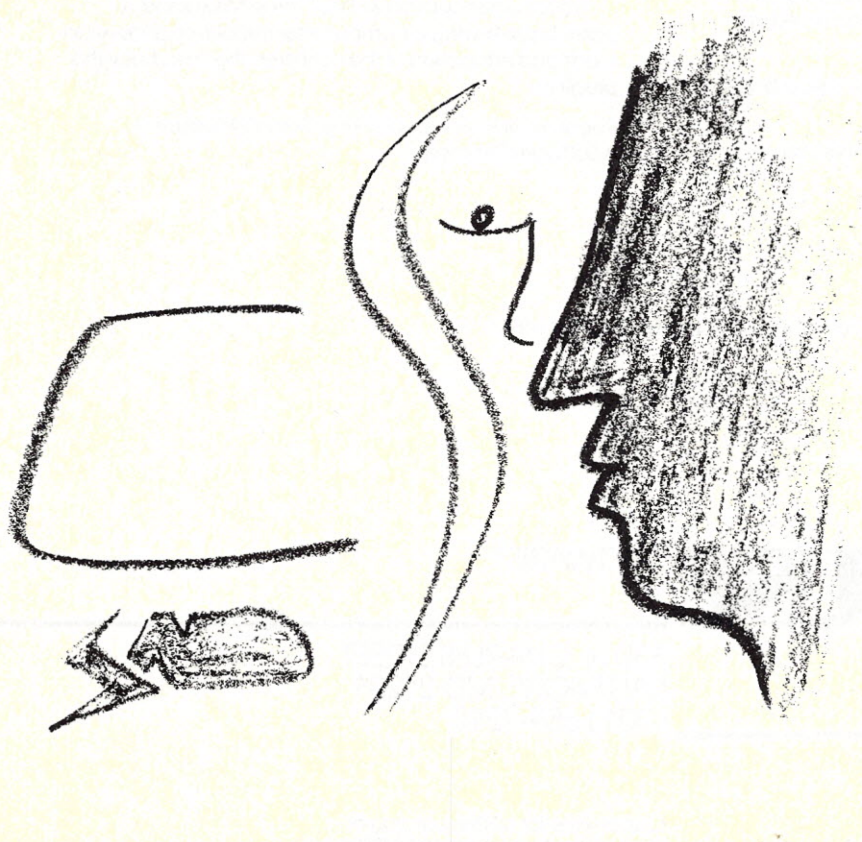
Despite pioneering work by Douglas Engelbart and others, popular opinion has it that "word processing" originated about 1967 with the appearance of IBM's magnetic tape typewriter.

Since then "word processing systems" have become epidemic. These are almost invariably disguised computers with a fixed program that acts only as an interactive text editor — you aren't allowed to run any other program. (Indeed, the salesman will deny ardently that it is a computer).

If you have a personal computer, however, the term's meaning shifts: a "word processor" becomes an interactive text-editing program that you run on your computer. (The marketing of personal computers is already beginning to impinge on the field of fixed-function word processors — the kind that won't let you run any other programs — and this is due to increase.)

Now, for some reason every technical wonk in the world thinks he is capable of writing a good word processing program. But in my opinion there does not exist a single satisfactory word processing program anywhere in the world.

This is not the place to hold forth the argument in detail. My basic view, however, is that a proper word processor should be, like any other interactive system, an artfully constructed virtuality, an integrated system of world, views and controls that is extremely easy to visualize, roam in and change.



Watch from afar, or enter the person's mind. You choose what happens.

Virtuality, cont'd...

Computer people (who are not usually concerned writers) have gotten some fairly twisted notions about the nature of text. The correct units of text are the *word, sentence, paragraph, heading* and *chapter.* Yet for some ungodly reason, computer people have gotten the idea that the units of text are the *character* (including control characters) and the *line* (or *linefeed).* This in turn leads to the unfortunate writers being told by the programmers that they're not thinking logically because they can't keep their minds on the invisible control characters. The programmers' heads have been in the wrong place, implementing the wrong virtuality.

(We will not even discuss here problems of the loss and protection of files, or the atrocity of short filenames required by some popular operating systems.)

Now, a case can be made for baroque word-processing programs that take weeks and months to learn, and allow you to format output in columns, windows and whatnot, but the fundamental problem of word processing is *fast input and revision,* and making the system easy to learn for everyone in the office (including office temps). Thus I think we need simplicity more than we need the baroque.

I have personally been designing word processing programs since 1960, and indeed the theory of virtuality presented here has co-evolved in part with the designs. But I will only inflict one of my text designs on the reader of this article, and that only loosely.

That design is the JOT™ system (Juggler of Text). And its special virtue is its ease of learning and use. In all cases where the system was demonstrated to a fresh user, the user learned to insert, delete and rearrange text in *under seven minutes.* (Unfortunately the implementation died, being in an obscure language on a now-defunct machine, and this figure is for a total number of less than twenty people. However, the system is presently being re-implemented for the Apple.)

Basic Operations

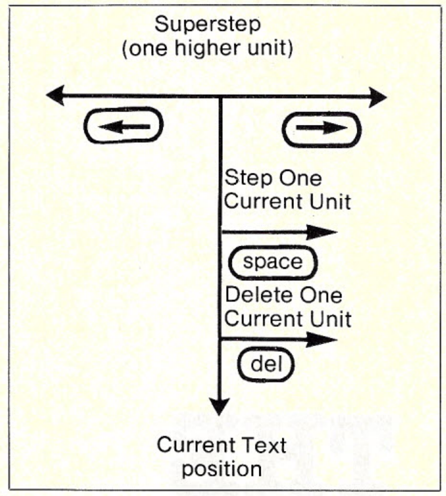
The basic operations of a simple word processor are essentially insert; step forward and back by word, sentence or paragraph; delete units of various sizes; and rearrange.

The JOT system was originally designed to do all this with the Teletype Model 33 as a terminal, capturing full upper and lower case even though that device has no shift key.

There are no command input strings. There are also no menus, a certain minimalist aesthetic having taken over. Half a dozen or so keys have been given new meanings which appear to hang together psychologically.

For instance, in one of the principal modes, left arrow means "move leftward to the beginning of the sentence" and the right arrow means "move rightward to the end of the sentence." The space bar means "step one word."

In the other main mode, the same controls mean "step left to the beginning of the paragraph," "step right to the beginning of the paragraph," and "skip one sentence."



I will not attempt to justify these commands in isolation; it is the whole system whose virtuality we design, and the individual commands stand or fall as part of this whole.

Some programmers consider the command structure of JOT to be detestable and illogical. But the system was not designed for programmers. It is intended to be usable for hours on end by non-computer people who are tense (sometimes frantic), and utterly preoccupied with the words, not with the machine.

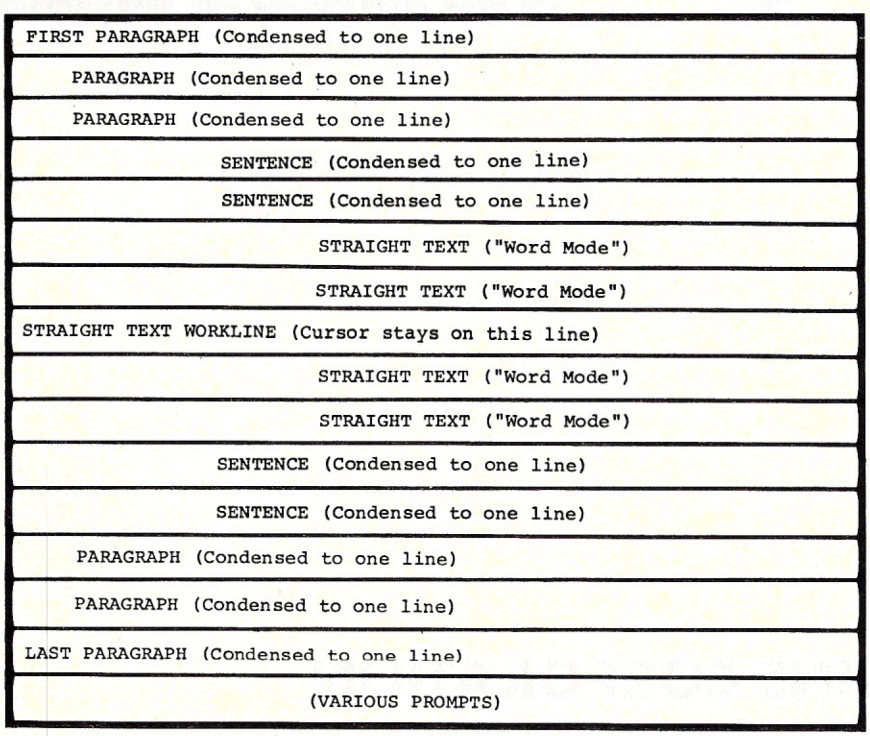
The actual state-diagram flowchart of the system is a hairy mess. (Indeed, the two programmers who implemented it originally agreed between themselves that it was all wrong — until they saw it in operation. Both became ardent virutality-design freaks. The fact that two exceptionally talented individuals were not able to imagine the system's performance from the flowchart shows what we are up against in general.)

I have spent no fewer than five hundred man-hours on the design of the JOT system's virtuality *only,* not counting any implementation. My impression is that most designers spend little or no time on virtuality design. While I make no claim that quality of work is ever proportional to the time spent, I think that the important thing is to design the interactive qualities first, *then* implement.

In essence the design process for JOT involved considering many different desirable features and operations, then cutting them down judiciously to a very small but powerful set that could be easily learned. In addition, it became a challenge to marry the desired control functions to a sparse keyboard (such as that of the Teletype or Apple) while still making it seem natural. Others can judge from the live system whether this has succeeded.

Regardless of the success of this particular design, we will return to these principles, and try to generalize them, later on. □

(TO BE CONCLUDED IN THE DECEMBER ISSUE)



ScreenJot™ — the expanded version of JOT — shrinks an entire document to one screenful, condensing sentences and paragraphs to one line each for orientation and overview. The cursor stays on the midline.

COMPUTRONICS MATHEMATICAL APPLICATIONS SERVICE™

EVERYTHING FOR YOUR APPLE* II

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

* **APPLE** IS A TRADEMARK OF THE APPLE CORP.

★ *All orders processed within 24-Hours*
★ *30-Day money back guarantee on all Software (less a $3 penalty for handling)*

FROM COMPUMAX BUSINESS SYSTEMS

The COMPUMAX business applications programs are written with the novice computer user in mind. They are easy to use, yet powerful in their capabilities. Further, COMPUMAX supplies the BASIC source code. Thus the programs are easy to modify.

MICROLEDGER

This General Ledger system performs the essential functions of dual entry bookkeeping and matches revenues and expenses:
MICROLEDGER includes the following programs.:
LEDGER 1 - builds and maintains the CHART OF ACCOUNTS file. This file contains both current and accumulated totals for each account.
LEDGER 2 - builds and updates the JOURNAL TRANSACTION file.
LEDGER 3 - lists both the the JOURNAL file and the CHART OF ACCOUNTS.
LEDGER 4 - computes the TRIAL BALANCE and executes POSTING of journal transactions into the CHART OF ACCOUNTS. An AUDIT TRIAL of all transaction is output.
LEDGER 5 - produces the PROFIT AND LOSS STATEMENT.
LEDGER 6 - produces the BALANCE SHEET. Assets, liabilities and owners' equities are shown by account and by totals. **$140.00**

MICROPAY

An Accounts Payable system, MICROPAY includes the following program & functions:
PAY 1 - initializes both Transaction and Master files, then begins the Accounts Payable process by inputting and adding records in the Transaction file.
PAY 2 - allows for changes and deletions of Transaction and Master records.
PAY 3 - reports outstanding Accounts Payables in four categories; under 30 days, 31-60 days, 61-90 days, and over 90 days.
PAY 4 - reports all outstanding Accounts Payables for a single customer or for all customers, and computes Cash Requirements.
PAY 5 - reports all outstanding Accounts Payables for a single date or for a range of dates and computes the Cash Requirements.
PAY 6 - lists both the Transactions and Master files.
PAY 7 - prints checks and accumulates and journalizes Accounts Payables. This program simultaneously creates entries for the MICROLEDGER file. **$140.00**

MICROREC

An Accounts Receivable system, MICROREC includes the following programs and functions:
REC 1 - initializes Accounts Receivable files, adds A/R record and prints invoices.
REC 2 - accepts receipt of customer payments and changes or deletions of A/R Transaction or Master file records.
REC 3 - reports outstanding Accounts Receivables in four categories; under 30 days, 31-60 days, 61-90days, and over 90 days.
REC 4 - reports all outstanding Accounts Receivables for a single customer, or for all customers and computes Cash Projections.
REC 5 - produces reports for all outstanding Accounts Receivables for a single date or for a range of dates and computes Cash projections.
REC 6 - lists Transaction and Master files and accumulates and journalizes Accounts Receivables, creating JOURNAL entries which communicate with the MICROLEDGER JOURNAL file. **$140.00**

MICROINV

This Inventory Control system presents a general method of Inventory Control and produces several important reports. Its program includes:
INV 1 - initializes Transaction and Master files and adds and updates Transaction and Master records.
INV 2 - handles inventory issued or received, creating inventory records. This program also accumulates and journalizes transactions, producing JOURNAL entries which communicate with the MICROLEDGER file.
INV 3 - lists both Transaction and Master files.
INV 4 - produces the STOCK STATUS REPORT, showing the standard inventory stock data and stock valuation, and the ABC ANALYSIS breaking down the inventory into groups by frequency of usage.
INV 5 - gives a JOB COST REPORT/MATERIALS, showing allocation of materials used year-to-date by each job or work code. (This is complemented by the Job Cost Report/Personnel in the MICROPERS program.)
INV 6 - computes and provides the E.O.Q. (Economic Order Quantities) **$140.00**

MICROPERS

This is a Payroll/Personnel program whose functions include:
PERS 1 - initializes the Master file and allows for entry and updates of Master records.
PERS 2 - initializes the Payroll file and allows for entry and updates of payroll records.
PERS 3 - lists an Employee Master Record or the entire Employee Master file; lists a single Payroll Record or the entire Payroll file.
PERS 4 - computes Payroll and prints the PAYROLL REGISTER. Prints PAYCHECKS and creates JOURNAL entries to be fed into the MICROLEDGER JOURNAL file.
PERS 5 - produces the JOB COST REPORT/PERSONNEL, computes the quarterly 941 bank deposit, and the Annual W-2 run. **$140.00**

All COMPUMAX programs available in machine readable format (diskette form) for the following machines:

TRS-80™ Model I	Micropolis 1053/11
APPLE II	Microsoft under CP/M
PET	CBASIC under CP/M
	Cromemco

FROM ADVENTURE INTERNATIONAL (By Scott Adams)

† 1. **ADVENTURELAND** - You wander through an enchanted world trying to recover the 13 lost treasures. You'll encounter wild animals, magical beings, and many other perils and puzzles. Can you rescue the Blue Ox from the quicksand? Or find your way out of the maze of pits? Happy Adventuring .....

† 2. **PIRATE'S ADVENTURE** - "Yo ho ho and a bottle of rum" You'll meet up with the pirate and his daffy bird along with many strange sights as you attempt to go from your London flat to Treasure Island. Can you recover Long John Silver's lost treasures? Happy Sailing, matey .....

3. **MISSION IMPOSSIBLE ADVENTURE** - Good morning, your mission is to...and so it starts. Will you be able to complete your mission in time? Or is the world's first automated nuclear reactor doomed? This one's well named. It's hard, there is no magic, but plenty of suspense. Good luck .....

4. **VOODOO CASTLE** - Count Cristo has had a fiendish curse put on him by his enemies. There he lies, with you his only hope. Will you be able to rescue him or is he forever doomed? Beware the Voodoo Man .....

5. **THE COUNT** - You wake up in a large brass bed in a castle somewhere in Transylvani. Who are you, what are you doing here, and WHY did the postman deliver a bottle of blood? You'll love this Adventure, in fact, you might say it's Love at First Byte .....

6. **STRANGE ODYSSEY** - Marooned at the edge of the galaxy, you've stumbled on the ruins of an ancient alien civilization complete with fabulous treasures and unearthly technologies. Can you collect the treasures and return or will you end up marooned forever? .....

7. **MYSTERY FUN HOUSE** - Can you find your way completely through the strangest Fun House in existence, or will you always be kicked out when the park closes? ...

8. **PYRAMID OF DOOM** - An Egyptian Treasure Hunt leads you into the dark recesses of a recently uncovered Pyramid. Will you recover all the treasures or more likely will you join its denizens for that long eternal sleep? .....

9. **GHOST TOWN** - Explore a deserted western mining town in search of 13 treasures From rattlesnakes to runaway horses, this Adventure's got them all! Just remember, Pardner, they don't call them Ghost Towns for nothin'. (Also includes new bonus scoring system!) .... **$14.95 Per Adventure**

* **Note:** Apple requires 24K and has no lower case.
† Recommended for the novice adventurer, with many built-in HELPS!

FROM PERSONAL SOFTWARE INC.

VISICALC **$150.00**

Take virtually any problem you would explore using calculator, pen, and paper, working in rows and columns. Apply VisiCalc and you'll see why every reviewer of this product has said the same thing: VisiCalc is the most useful, most important program yet developed for personal computing.

With VisiCalc, you work with an electronic worksheet of up to 63 columns and 254 rows. At the juncture of any column and row you can type in words and numbers. VisiCalc automatically performs all arithmetic functions, net present value, and transcendental functions - instantly!

CCA DATA MANAGEMENT SYSTEM $74.95

DMS Features:
File Creaton and Maintenance:
- Fields may be alphanumeric, numeric, integer, floating point, or fixed decimal with commas.
- Fields may be COMPUTED FIELDS. DMS will compute any field within a record, using constants or other fields in the same record. Functions include add, subtract, multiply, divide, and raise exponential powers.
- Records are easily located, using the SCAN feature. SCAN for records with a field over, below, or between a range of values.
- Records are easily added and updated. DMS "prompts" you with questions.
- Multi-diskette capabilities for larger files - up to 85,000 characters per file!
- Sort the records into almost any order, using up to 10 fields as "keys". So you can sort for customer numbers; within zip code, for instance.
- Delete records, "compact" files, and backup files on data diskettes easily.

Report Features:
- Print reports with records in any order.
- Select fields to be printed.
- Print mailing labels.
- Numeric totals and subtotals can be specified when a value in an unrelated field in the same record changes. For example, sort, subtotal, and print according to department, or month, or customer number, or model number.

GUARANTEED PROFIT **91%** WINS PLACES SHOWS **32%** AVERAGE PROFIT AT ALL TRACKS-1978

THE HORSE SELECTOR II (FLATS) (By Dr. Hal Davis) $50.00

New simplified version of the original Horse Selector. The first Horse Selection System to actually calculate the estimated odds of each horse.
HIGHER PROFITS (OVER 100%) POSSIBLE THROUGH SELECTIVE BETTING ON:
- Rates each horse in 10 seconds.
- Easy to follow rules.
- Can be used with any Apple II Computer.
- 100% money back guarantee (returned for any reason).
- Uses 4 factors (speed rating, track variant, distance of the present race, distance of the last race)
- Using the above factors, the Horse Selector calculates the estimated odds. BET on horses whose actual payoff (from the Tote Board or Morning Lines) is higher than payoff based on estimated odds.
- Using the above factors, the Horse Selector calculates the estimated odds. BET on any selected horse with an estimated payoff (based on Tote Board or Morning Lines) higher than calculated payoff (based on Horse Selector II).
- Source listing for the TRS-80™, TI-59, HP-67, HP-41, Apple and BASIC Computers.
- No computer or calculator necessary (although a calculator would be helpful for the simple division used to calculate estimated odds)

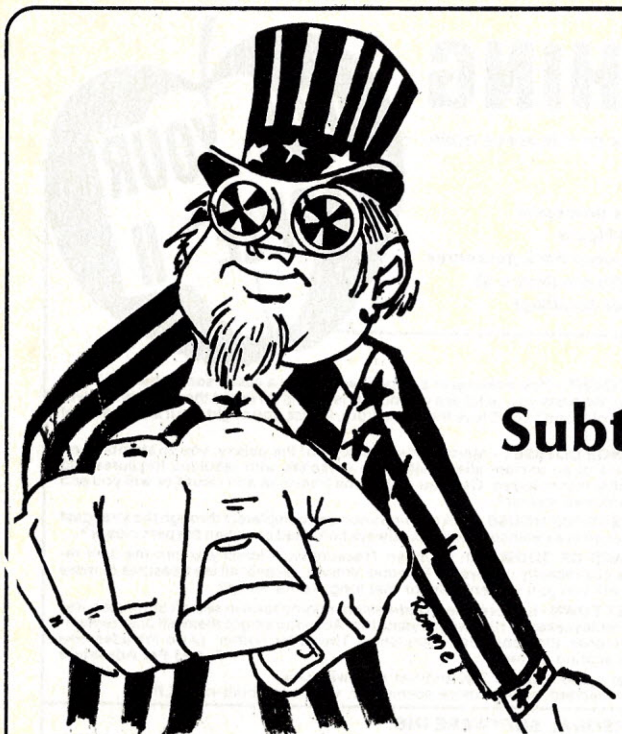
FREE Dutching Tables allows betting on 2 or more horses with a guaranteed profit.

master charge **24 HOUR ORDER LINE** VISA
(914) 425-1535

ADD $2.00 FOR SHIPPING UPS AREAS
ADD $3.00 FOR C.O.D. OR NON-UPS AREAS
ADD $4.00 OUTSIDE U.S.A, CANADA & MEXICO

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

CIRCLE 138 ON READER SERVICE CARD



Pleas, Prayers and Subtle Hints for Programmers

McLaughlin and Lubar

This topic has been covered before, and will probably be covered again, but an ease-of-use issue seemed the perfect opportunity to broach the subject. The point of all this is to save everyone a bit of time and anguish. We like to get software submissions, but there are certain guidelines that should be followed.

First, there is the hideous beast known as "commercial potential." Ours not being an ideal society, crass considerations such as this must be discussed. To put it bluntly, if a program has no chance of selling, we just can't go through the expense of producing and marketing it, no matter how well done or aesthetically pleasing it is. There is a simple test that will help determine a product's marketability. Ask yourself if you would lay out hard-earned cash for the program. And going one step further, honestly consider how many others might do the same.

What makes a program unmarketable? The most obvious reason would probably be a lack of originality. Haven't we seen enough versions of blackjack, tic-tac-toe and Star-trek? This should not be confused with games which are old to the world but new to the computer. For example, pinochle is an old game, but there are no commercially available programs that play this game. Also, the existence of a program does not eliminate the possibility of a truly improved or enhanced version. Just remember that this means a major modification, not a slightly changed rework of the same old thing.

Secondly, the program must be written to run on a popular machine with no special hardware required. Being realistic (and, alas, we must), you cannot expect to sell a thousand copies of a program if only three hundred people own a machine on which it will run. Here at *Creative,* we are currently looking for software that runs on the Apple II, Atari and TRS-80 Level II computers. While others may be considered, and the list is certainly open to revision, at the moment this is where we see the largest market.

If the program is a game (not all are, you know), you must also consider its level of challenge. People, not liking to be either bored or frustrated, do not want to spend money on a game that they can always win

Haven't we seen enough versions of blackjack, tic-tac-toe and Star-trek?

or on one that they can never win. With this in mind, we realize that providing multiple levels of difficulty will make the game appeal to a wider audience. Another thing to remember is that if the computer is to be an opponent, it must play a very good game but must not cheat (yes, we have seen dishonest programs).

O.K., so, in concept, you have a marketable program. Now what? For one thing, the instructions must be understandable. Some games, such as chess, have complex rules, but it is possible to present these rules in such a way that the average person can grasp them. Interposing demonstrations with rules is one good approach, and it helps break up the tedium of large blocks of text. Of course, the user should always have the option of skipping the instructions.

Next, the program should be clean. That means, not only should it run without getting any SYNTAX ERRORs, but should do extensive error checking so that even the most novice user cannot suddenly find himself in the mysterious world of Basic. Input prompts should be understandable (as opposed to cryptic). And if a number from 1 to 4 is requested, the program should not accept a 7 or B (and then make an absurd assumption about it). By the way, correct spelling would certainly be a nice touch.

If appropriate, a program should use the full capabilities of the computer. The day of teletype graphics is happily past. Does your machine have color or sound? Good, put them to use to make your game more exciting and fun or to make your application easier to use or less prone to user error.

The above discussion concerns the game itself. There are some other considerations dealing with the actual submission. The program must be in a machine readable form; namely tape or disc. A number of submitters fail to mention whether they are sending the program for software or magazine publication. Some even neglect to mention which computer the program is written for. Needless to say, this does not contribute to a good first impression. Documentation should also be included, explaining anything a user would need to know. Any system-specific distinctions should be made; new or old ROMS for the PET, Level I or II for the TRS-80, and so on.

If you want your program returned, an envelope with return address and postage is a must. That about covers it. Every program submitted gets careful consideration, and we expect to see a lot of good material coming from you folks out there. So get to it. □

Now NRI takes you inside the world's most popular microcomputer to train you at home as the new breed of computer specialist!

NRI teams up with Radio Shack to teach you how to use, program and service microcomputers...make you the complete technician.

It's no longer enough to be just a programmer or a technician. With microcomputers moving into the fabric of our lives (over 200,000 of the TRS-80™ alone have been sold), interdisciplinary skills are demanded. And NRI can prepare you with the first course of its kind, covering the complete world of the microcomputer.

Learn At Home in Your Spare Time

With NRI training, the programmer gains practical knowledge of hardware, enabling him to design simpler, more effective programs. And, with advanced programming skills, the technician can test and debug systems quickly and easily.

Only NRI gives you both kinds of training with the convenience of home study. No classroom pressures, no night school, no gasoline wasted. You learn at your convenience, at your own pace. Yet you're always backed by the NRI staff and your instructor, answering questions, giving you guidance, and helping you over the tough spots.

Training includes TRS-80 computer, transistorized volt-ohm meter, digital frequency counter, and the NRI Discovery Lab with hundreds of tests and experiments.

(TRS-80 is a trademark of the Radio Shack division of Tandy Corp.)

Explore the TRS-80 Inside and Out

NRI training is hands-on training, with practical experiments and demonstrations as the very foundation of your knowledge. You don't just program your computer, you introduce and correct faults ...watch how circuits interact...interface with other systems...gain a real insight into its nature.

You also build test instruments and the NRI Discovery Lab, performing over 60 separate experiments in the process. You learn how your trouble-shooting tools work, and gain greater understanding of the information they give you. Both microcomputer and equipment come as part of your training for you to use and keep.

Send for Free Catalog... No Salesman Will Call

Get all the details on this exciting course in NRI's free, 100-page catalog. It shows all equipment, lesson outlines, and facts on other electronics courses such as Complete Communications with CB, TV and Audio, Digital Electronics, and more. Send today, no salesman will ever bother you. Keep up with the latest technology as you learn on the world's most popular computer. If coupon has been used, write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016
NO SALESMAN WILL CALL
Please check for one free catalog only.

All career courses approved under GI Bill.
☐ Check for details.

- ☐ Computer Electronics Including Microcomputers
- ☐ TV/Audio/Video Systems Servicing
- ☐ Complete Communications Electronics with CB • FCC Licenses • Aircraft, Mobile, Marine Electronics
- ☐ CB Specialists Course
- ☐ Digital Electronics • Electronic Technology • Basic Electronics
- ☐ Small Engine Repair
- ☐ Electrical Appliance Servicing
- ☐ Automotive Mechanics
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Refrigeration, & Heating including Solar Technology

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council

CIRCLE 214 ON READER SERVICE CARD

How to Solve It — With the Computer

Part Three

Donald T. Piele

> *"Leibniz saw in his binary arithmetic the image of creation . . . He imagined that unity (1) represented God, and zero (0) the void; that the Supreme Being drew all beings from the void, just as 1 and 0 express all numbers in his system of numeration."*
>
> *Tobias Dantzig, 1939*

The binary number system lies at the heart of all computer operations. In this system all numbers can be expressed by a finite set of 0's and 1's which can be easily represented by a low and high voltage. Other systems, easier for humans to handle, are the decimal and hexadecimal. These systems use more symbols for each digit and consequently the same number can be expressed in a shorter length. For example, the number 111 (binary) is equivalent to 15 (decimal) and F (hexadecimal).

All three systems are used in connection with computing. Thus to be computer literate, one should understand the interrelationship of these three number systems. This months selection of problems for intermediate and advanced students will explore number systems and related problems. For the beginning student I will continue last month's exploration of Apple graphics and introduce FOR-NEXT loops.

Lesson #3 (Beginning Students)
Apple Graphics

In the previous two lessons, I introduced the graphics commands GR, COLOR, PLOT A,B and END. I showed how the program

```
10 GR                        40 COLOR=C
20 INPUT "COLOR=",C          50 PLOT X,Y
30 INPUT " X,Y =",X,Y        60 END
```

will place a small colored square at the position X,Y on the 40x40 low resolution graphics screen. Thus, any one of the 1600 positions on the screen may be colored with the PLOT command; but will it take 1600 individually typed PLOT statements to cover the screen a specific color? At the rate beginning students type, it would take two weeks to write such a program. A typical solution might begin as:

```
20 COLOR=9
30 PLOT 0,0 : PLOT 1,0 : PLOT 2,0
40 PLOT 3,0 : PLOT 4,0 : PLOT 5,0
........

5360 PLOT 39,39
5370 END
```

and end after 537 lines. Clearly, the time is ripe to introduce a statement which will automate this process.

FOR-NEXT Loops

I prefer to introduce a new statement with a problem. The purpose of this problem is twofold-to introduce the new command FOR-NEXT and to exercise the problem solving process.

Donald Piele, Associate Professor of Mathematics, University of Wisconsin-Parkside, Kenosha, WI 53141.

Problem #3 (Beginning Students)
Write a small program (less than 10 statements) that will fill the screen with a single color.

Discussion:

There are two similar ways to view this problem. One strategy is to first color the positions in a given row left to right, and then to repeat the procedure for each row from top to bottom. The other strategy is to begin by coloring a column from top to bottom and then to repeat the procedure for each column from left to right. Both strategies involve the use of subgoals. Using the first strategy, the subgoal is to create a procedure for coloring a row. This can be done with a FOR-NEXT loop as follows.

```
          C O L U M N

        0 1 2 3 4 .....39
     0  ________________     10 GR
  R  1                       20 COLOR=9
  O  2                       30   FOR I=0 TO 39
  W  .                       40   PLOT I,0
     .                       50   NEXT I
    39                       60 END
```

I usually key in this program, run it, and begin asking questions.

1. List all the points that are plotted. (0,0 1,0 ....38,0 39,0)
2. How can I change the program to color the second row? (40 PLOT I,1)
3. How can I change the program to color the last row? (40 PLOT I,39)
4. List the diagonal positions from the upper left corner to the bottom right corner. (0,0 1,1 2,2 ....39,39)
5. How can I change line 40 to draw the diagonal for me? (40 PLOT I,I)
6. List the positions in the first column. (0,0 0,1 ...0,39)
7. How can I change the program to color the first column? (40 PLOT 0,1)
8. How can I change line 40 to color the last column? (40 PLOT 39,I)

When I began teaching Basic to third and fourth graders, I wondered if the use of variables, such as (I) in the above program, would be too abstract to be easily understood by young minds. My fears were quickly laid to rest as my students began to use variables with ease almost immediately. In fact, experiences with older students and adults have convinced me that the younger ones pick up the use of variables faster. I no longer keep any secrets from the younger set.

There are two commands specific to the Apple II that make drawing horizontal and vertical lines even easier. These are HLIN and VLIN. To color the first row orange simply write:

```
10 GR
20 COLOR=9
30 HLIN 0,39 AT 0
40 END
```

Similar questions and modifications can be made to this program.

1. How can I change the program to color the last row?
(30 HLIN 0,39 AT 39)
2. How can I change the program to color the first column?
(30 VLIN 0,39 AT 0)
3. How can I change the program to color the last column?
(30 VLIN 0,39 AT 39)

HLIN and VLIN can be used to construct horizontal and vertical lines of length (1-40) anywhere on the screen. For example, to draw a line in row 3 beginning in column 5 and ending in column 12 write
30 HLIN 5,12 AT 3.

The following two programs illustrate two ways of constructing this line.

```
10 GR                    10 GR
20 COLOR=9               20 COLOR=9
30 FOR I=5 TO 12         30 HLIN 5,12 AT 3
40 PLOT I,3              40 END
50 NEXT I
60 END
```

```
              C O L U M N

      0 1 2 3 4 5 6 7 8 9 10 11 12 .....39
    0
R   1
O   2
W   3           ________________
    4
    .
    .
   39
```

What changes must be made to the above two programs to produce a vertical line in column 3 beginning at row 5 and ending at row 12?

```
[ 40 PLOT 3,I              30 VLIN 5,12 AT 3 ]
```

After giving the students what I consider to be enough experience with constructing lines, which are subgoals for problem #3, I prefer to say, "no more" and let the students go to work on the problem. Now they are the ones who will be asking questions — as it should be. I am available to answer their questions about all commands and statements, but it is up to them to experiment putting them together to solve the problem. Here are two ways that it could be done.

```
10 GR                    10 GR
20 COLOR=9               20 COLOR=9
30 FOR I=0 TO 39         30 FOR J=0 TO 39
40    HLIN 0,39 AT I     40    FOR I=0 TO 9
50 NEXT I                50       PLOT I,J
60 END                   60    NEXT I
                         70 NEXT J
                         80 END
```

These two programs use the strategy of covering the screen one row at a time. The same programs can be used to cover the screen one column at a time with the following changes:

```
[ 40 VLIN 0,39 AT I           50 PLOT J,I ]
```

Intermediate Students

The idea of representing natural numbers in different bases is fundamental in computer programming. The binary, decimal, and hexadecimal number system are introduced in all elementary assembly language programming books and in many Basic programming books. At the same time, the study of bases has diminished in mathematics textbooks along with other "new math" concepts. Lacking any immediate application and given the confusion it caused parents, the study of bases has become unpopular.

But the idea of bases is fundamental in computer programming and can be introduced to young students with a few matchboxes.

```
 |__|      |__|      |__|      |__|
Box 4     Box 3     Box 2     Box 1
```

Solve, cont'd. . .

Suppose each box has the capacity to hold B matches. B is called the base for the number system and is usually expressed as a decimal number. To express the numbers in base B in sequential order, begin by adding a match one at a time to box 1. As soon as any match box is filled (contains B matches) empty it and add one match to the next higher box on its left. To represent the number of matches in each box use the following symbols:

```
Number of
  matches =   0 1 2 3 4 5 6 7 8 9 10 11 12 ....... 36

Single symbol  =   0 1 2 3 4 5 6 7 8 9  A  B  C     Z
```

Each number is expressed in base B by writing the sequence of symbols associated with the number of matches in each box. For example, if the base B = 3, then the natural numbers would begin as 0001, 0002, and change to 0010 as the first box filled up with 3 matches, was emptied, and one match was deposited in the next higher box (#2).

It is an instructive programming exercise to implement this procedure on the computer. Arrays come in handy for storing the number of matches in each box:

B (I)= # of matches in Box I,

and string arrays are useful for holding the symbols;

```
S$="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ".
```

(I have limited the size of the base to a maximum of 36 for practical reasons.)

Problem #3 (Intermediate Students)

Write a program that displays the natural numbers in base B in sequential order up to N digits.

Plan of Attack

Using the match box simulation, begin by depositing one match in box 1;

100 B(1)=B(1)+1.

Next check the number of matches in box 1 and continuing adding matches to box 1 if it is not filled.

IF B(1) <B THEN (print number and GOTO 100)

As soon as the first box fills up, empty it and add one match to the next higher box (#2).

B(1)=0
B(2)=B(2)+1

Continue adding matches to box 1.

IF B(2) < B THEN (print number and GOTO 100)

Otherwise, empty the matches in box 2 and add one to box 3.

B(2)=0
B(3)=B(3)+1

This same procedure is continued until a match finally reaches box N+1.

Here is a sample solution that implements this procedure.

```
     First Sample Solution

 10 DIM S$(36)
 20 S$="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
 30 INPUT "BASE =",B
 40 INPUT "N PLACES = ",N
 50    REM **** GENERATE THE SEQUENCE OF DIGIT****
 60 B(1)=B(1)+1
 70 FOR I=1 TO N
 80    IF B(I) < B THEN EXIT 130
 90       B(I)=0
100       B(I+1)=B(I+1) + 1
110 NEXT I
120 IF B(N+1)>0 THEN END
130    REM **** PRINT OUT ROUTINE ****
140 FOR I=N TO 1 STEP -1
150    PRINT S$( B(I)+1 , B(I)+1 ),
160 NEXT I
170    PRINT
180       GOTO 50
190 END
```

Remarks

1. For Microsoft Basic change S$(B(I)+1,B(I)+1), in line 150 to MID$(S$,B(I)+1,1);
2. It is advisble to have the students add documentation at the beginning of their program to explain what the program is doing.

For example, I require the following:

```
1 REM ************* PROBLEM #3 **********
2 REM                 BY DON PIELE
3 REM                 AUGUST 28, 1980
6 PRINT "THIS PROGRAM WILL GENERATE AND PRINT THE"
7 PRINT "NATURAL NUMBERS IN BASE B UP TO N PLACES."
```

Another way to attack problem #3 is to use a coversion algorithm from the base 10 to the desired base B. To convert the number K into a base B number, first divide by B and save the quotient (INT (K/B) and the remainder (K — B*INT(K/B)). The remainder represents the value of the first "digit" (B(1)) and the quotient becomes the new number K. The process is repeated until K=0. For example, to convert 53 into base 5,

53/5= 10 remainder of 3 (=B(1))
10/5= 2 remainder of 0 (=B(2))
2/5= 0 remainder of 2 (=B(3))
Thus 53 (base 10)= 203 (base 5).

This procedure can be implemented by changing lines 60-120 in the first sample solution and adding line 175.

```
   Program Modification for Second Solution
   ........................
   60 K=C
   70 FOR I=1 TO N
   80     B(I)=K-B*INT(K/B)
   90       K =INT(K/B)
  100 NEXT I
  110 REM
  120 IF K>0 THEN END
   .........................
  175 C=C+1
```

Related Ideas

An unusual method for multiplying two numbers, said to be in common use in Russian villages around the turn of the century, is known today as the Russian Peasant Method. The only skills one needs to multiply with this method is to be able to add and divide by two. A description of the procedure is given below as it appeared in 1912 in *The Mathematics Teacher.*

"Having given the positive integers A and B, to multiply A by B write down A x B; under A write the exact or lower quotient obtained by dividing by 2 (INT (A/2)); under this quotient write the exact or lower quotient obtained by dividing by 2, and so on until you obtain the quotient 1. Under B write its double, under this double its double, and so on, until you have as many numbers in the second column as in the first. Next add the numbers in the second column which correspond to odd numbers in the first column. The result is the product of A x B."

Below is a specific example using 21 x 13 which illustrates the technique.

```
                 A        B
                ----------
                21   X   13     *  |
   Divide  |    10       26        |
     by    |     5       52     *  |  Double
    Two    v     2      104        v
                 1      208     *
                -------------
                21 x 13 = 13 + 52 + 208 = 273
```

The product (21 x 13)9 is equal to the sum of all the values in column B (indicated with a *) which correspond to odd values in column A.

Solve, cont'd. . .

Russian Peasant Problem (Intermediate Students)

Write a program that uses the Russian Peasant Method to multiply two positive integers. Explain why the method works.

```
Sample Solution (Russian Peasant Problem)

1 REM **** RUSSIAN PEASANT METHOD OF MULT ****
 10 S=0
 20 INPUT " A , B = ",A,B
 30 X=A
 40 Y=B
 50 IF X/2 <> INT(X/2) THEN S=S+Y
 60 X=INT(X/2)
 70 Y=2*Y
 80 IF X >= 1 THEN 50
 90 PRINT  A," x",B," =",S
100 END
```

Note

A justification for the Russian Peasant Method of multiplication can be given by:
(1) expressing 4 in a base 2;
21 (base 10) = 10101 (base 2);
(2) using the appropriate powers of 2 corresponding to 10101 to write;
$21 = 2^4 + 2^2 + 1.$
Thus, $21 \times 13 = (16 + 4 + 1) \times 13 = 208 + 52 + 13 = 273.$

Advanced Students

The Russian Peasant Method can be used for numbers expressed in other bases besides 10. It can also be implemented for multiplying large positive integers when the digits are kept in an array. For example, if the product A x Z exceeds 8 digits (or higher depending on the Basic used), it is nesessary to store the numbers in arrays and perform array arithmetic. This suggests an extension of the previous problem to the following.

Problem #3 (Advanced Students)

Write a program that uses the Russian Peasant Method of multiplication to multiply numbers, up to 20 digits in length, in any base B.

Remarks

1. This problem reguires an extensive use of subgoals which can be implemented with subroutines or by defining functions.
2. The first task is to write a procedure which will transform any number into an array where each element of the array is written in base 10. Thus if A=123456789ABCDEF (base 16), then the corresponding array A(I) is constructed as follows,
A(1)=16, A(2)=15, A(3)=14, ... A(16)=1.
3. A subroutine must be written that will test whether A is even or odd with respect to the base B.
4. A subroutine must be written to double the number Z (in the array with respect to the base B.
5. A subroutine must be written to divide A (in array form) by 2 and test if A is less than 1.
6. A subroutine must be written to print out the sum of all Z arrays corresponding to odd A arrays consistent with the Russian Peasant Method.
7. Finally, it is advisable to make A the smallest number since successive divisions by 2 will reduce A to 1 more quickly. Thus, a procedure should be written to check the lengths of A and Z and to make A the smallest number if necessary.

Listed below is a sample solution to this problem. I have intentionally written this program with subroutines instead of multiple-lined functions because the latter are not commonly available on the small personal computer systems. Also, for the sake of clarity, I have not attempted to combine routines that share common structures to save a few lines of programming. I have restricted the base B to ≤ 10, but it could be expanded to 36 by changing the way the numbers are read into arrays and the way the product is printed out.

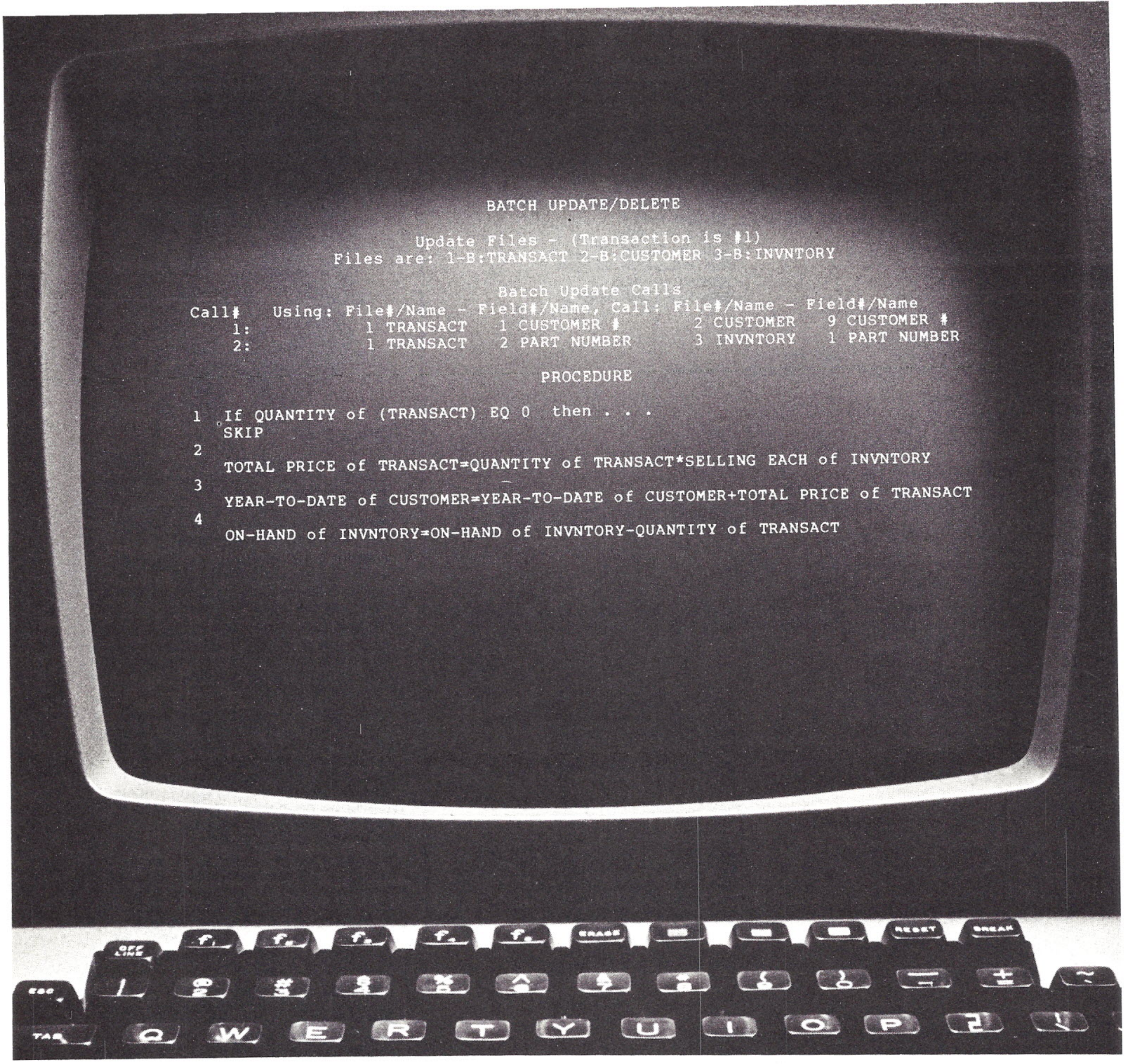
```
Sample Solution to Advanced Problem #3
10 PRINT "RUSSIAN PEASANT METHOD FOR MULTIPLICATION OF A x Z"
20 PRINT "A,Z LARGE WHOLE NUMBERS IN ANY BASE B <= 10 "
30 GOTO 360
40 REM  ***** SUBROUTINE TO TEST FOR EVEN OR ODD ******
50 C=0                                 \ REM C = CARRY
60 FOR I=N TO 1 STEP -1
70    X=A(I)+C
80    IF X/2 = INT(X/2) THEN C=0 ELSE C=B
90 NEXT I \ RETURN                    \ REM IF C>0 THEN A( ) IS ODD
100 REM  ***** SUBROUTINE TO ADD Z( ) TO SUM *****
110 C=0                               \ REM C = CARRY
120 IF M>K THEN K=M                   \ REM M=LENGTH OF Z( )
130 FOR I=1 TO K                      \ REM K=LENGTH OF SUM
140    X=S(I)+Z(I)+C                  \ REM S( )= SUM
150    R=X-B*INT(X/B)                 \ REM R = REMAINDER
160    S(I)=R   \ C = INT(X/B)
170 NEXT I
180 IF C>0 THEN S(K+1)=C
190 IF C>0 THEN K=K+1    \ RETURN
200 REM ***** SUBROUTINE TO DOUBLE Z( ) *****
210 C=0
220 FOR I=1 TO M
230    X=2*Z(I) + C
240    R=X-B*INT(X/B)
250    Z(I)=R    \ C = INT(X/B)
260 NEXT I
270 IF C>0 THEN Z(M+1)=C
280 IF C>0 THEN M=M+1    \ RETURN
290 REM ***** SUBROUTINE TO DIVIDE A( ) IN HALF (TO LOWEST INTEGER)
300 C=0 \ T=0                        \ REM T = TOTAL OF DIGITS IN A( )
310 FOR I=N TO 1 STEP -1
320    X=A(I)+C
330    IF X/2=INT(X/2) THEN C=0 ELSE C=B
340    A(I)=INT(X/2)    \ T=T+A(I)
350 NEXT I              \ RETURN
360 REM****************** M A I N   P R O G R A M ********************
370 DIM A$(20),Z$(20),T$(20),A(20),Z(40),S(40)
380 INPUT " BASE = ",B
390 INPUT "    A = ",A$
400 INPUT "    Z = ",Z$
410 N=LEN(A$) \ M=LEN(Z$)
420 IF N>M THEN K=N ELSE K=M          \ REM K = LARGEST LENGTH
430 IF M>=N THEN 460                  \ REM CHECK LENGTHS OF A AND Z
440 T$=A$ \ A$=Z$ \ Z$=T$             \ REM EXCHANGE A AND Z
450 GOTO 410
460 FOR I= 1 TO N                     \ REM PUT A$ INTO ARRAY A( )
470    A(I)=VAL(A$(N+1-I,N+1-I))
480 NEXT I
490 FOR I=1 TO M                      \ REM PUT Z$ INTO ARRAY Z( )
500    Z(I)=VAL(Z$(M+1-I,M+1-I))
510 NEXT I
520 GOSUB 40                          \ REM TEST A( ) FOR EVEN OR ODD
530 IF C>0 THEN GOSUB 100             \ REM IF ODD THEN ADD Z( ) TO SUM
540 GOSUB 200                         \ REM DOUBLE THE ARRAY Z( )
550 GOSUB 290                         \ REM DIVIDE A( ) IN HALF
560 IF T>0 THEN 520                   \ REM REPEAT UNTIL A( ) < 1
570 PRINT A$," x ",Z$, " = "          \ REM PRINT THE PROBLEM
580 FOR I=K TO 1 STEP -1
590    PRINT S(I),                    \ REM PRINT THE ANSWER
600 NEXT I    \ PRINT
610 END

                    S A M P L E   R U N

RUSSIAN PEASANT METHOD FOR MULTIPLICATION OF A x Z
A,Z LARGE WHOLE NUMBERS IN ANY BASE B <= 10

 BASE = 10
    A = 123456789
    Z = 12345678987654321
123456789 x 12345678987654321 =
 1 5 2 4 1 5 7 8 8 5 8 4 0 5 7 3 1 1 2 6 3 5 2 6 9
 BASE = 2
    A = 1010101010101010
    Z = 1010101010101010

1010101010101010 x 1010101010101010 =
 1 1 1 0 0 0 1 1 1 0 0 0 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 1 0 0
```

Remarks

1. In Microsoft Basic lines 470 and 500 need to changed using the equivalence A$(I,I) =MID$(A$,I,1).
2. To upgrade this program to handle any base B, the conversion routine (lines 470-510) and the output routine (lines 580-600) both need to be changed. □

References

1) Bowden, Joseph, "The Russian Peasant Method of Multiplication" *The Mathematics Teacher*, Vol. 5, page 4-8, 1912.
2) Dantzig, Tobias, *Number, The Language of Science*, Macmillan Publishing Company, New York, 1939.



The Ultimate Application Development System

Nothing can compete with the brain when it comes to information storage capacity and speed of data entry and recall — but we're working at it.

Our **SELECTOR-IV™** data base management system will let your microcomputer operate with the flexibility available (up to now) only on larger systems. You can create, maintain and report on files limited in size only by your *CP/M™ compatible operating system or disk storage capacity.

The basis of the power of **SELECTOR-IV™** is our unique method of cross-indexing the information in your files. You can immediately recall records by the contents of any piece of information required — from account numbers to ZIP codes to the date of your last audit. You can update records, individually or all at once. You can create new, uniquely, selected sub-files from existing ones (in the same or a different format), and perform computations in the process. You can define procedures to generate computed invoices, personalized letters, or gummed labels with the information coming from several files at once, and invoke them whenever needed. You can add new items to a record definition and change or delete them at will.

We've come a long way since we released the first information management system in microcomputers. We've listened to your suggestions and incorporated the best of them. We've built screen editing functions into the system which make operating the system as convenient as possible. We've had **SELECTOR-IV's™** documentation produced by our experts emphasizing its use for the novice, the applications developer, as well as, the retailer. Our applications specialists can provide you with a "turnkey" **SELECTOR-TV™** system customized for virtually any requirement.

With **SELECTOR-IV™** and a good word processor program, chances are you won't need any other software.

Look for SELECTOR-IV™ at your local computer retailer, or call:

MICRO•AP, INC.
9807 Davona Dr.
San Ramon, CA 94583
(415) 828-6697

MICRO•AP

*CP/M is a registered trademark of Digital Research.

CIRCLE 154 ON READER SERVICE CARD

Actors 2. The Story Continues.

Symposium On Actor Languages (Cont'd From October)

Brief Glossary of Actor - Related Terms

Actor (Hewitt) software entity whose behavior is predetermined by the class of which it is an example. Receives messages and sends them; these messages may in turn set other Actors in motion.

instance (Sutherland) original term for Actor.

object Actor.

class what an Actor is a member of.

class definition the behavioral specification, or response list, stating what Actors of a given class do. In Actor languages, the class definitions are the programs.

class editor program that allows you to change the class definition. In Actor languages, the only way to write the other programs is usually through a class editor.

superclass class which takes in another class: all the responses of the superclass are shared by an Actor in the class underneath.

inheritance the inclusion of a superclass' responses in its daughter class. If a superclass definition responds to message A with B, so do all Actors in classes beneath.

control structure the means of transferring control amongst different parts of a program; in conventional languages, such commands as GOTO, IF, and the subroutine jump.

flow of control the sequence in which a program is followed.

hair (hairiness, hairy) disorderliness of control structure, as in such languages as Act I.

demon (Selfridge) entity of predetermined behavior who responds to conditions of data.

sprite (Kornfeld) demon with certain special features.

workspace collection of programs and related data, stored all together as a unit.

environment workspace and/or operating system within which programs function.

Department Of Notational Engineering

That's what it says on the door of Drew McDermott, Yale professor of computer science.

It's good for a chuckle if you know something about the zany and abstract world of computer languages.

If you think more about it, it's very profound.

A computer language is a way of writing directions to a computer. A language *processor* — the compiler or interpreter that carries out the directions written in that way — is secondary to the notation itself, which represents a way of taking apart problems and putting together solutions.

To find a notation for something is to represent a structure. The notation permits manipulation and the growth of understanding. "Notational engineering" isn't a joke — it may be the rock bottom idea. □

Actor Language Bibliography

1. Classics

Ivan E. Sutherland, *Sketchpad: A Man-Machine Graphical Communication System.* Ph.D. thesis, MIT, 1963. (Excerpted in Proceedings of the Spring Joint Computer Conference, 1963.)

Alan Kay, *The Reactive Engine.* Ph.D. thesis, University of Utah, 1969.

Marvin Minsky, "Steps toward Artificial Intelligence." Proceedings of the Institute of Radio Engineers, January 1961.

Warren McCullough, *Embodiments of Mind.* MIT Press, 1965.

Allen Newell and Herbert Simon, *Human Problem Solving.* Prentice-Hall, 1962.

Ole-Johan Dahl and Kristen Nygaard, "SIMULA — an Algol-Based Simulation Language." *Communications of the ACM,* September 1966, 671-677.

Wallace Feurzeig et al., *Programming Languages as a Conceptual Framework for Teaching Mathematics: Final Report on BBN LOGO Project.* Bolt, Beranek and Newman (Boston), June 1971.

2. Actor-Language Concepts

Alan Kay, "FLEX, a Flexible Extensible Language." University of Utah Depart-

CONTINUED ON PG. 92

What—Or Who —Is "Artificial Intelligence"?

"Artificial intelligence" is no one thing. It is the cult — disguised as an academic field — of brilliant zanies who talk to each other about mental processes considered in the abstract. They talk and write about methods of search, reference, representation of the "real world" in data structures; about language design; about robots. They also program about these things.

This group refers frequently to the "real world" — perhaps more than any other group — but in a way that makes it seem like *somewhere else.*

You'll find them at MIT, Stanford, Carnegie-Mellon, Case, Utah, Caltech, PARC, and wherever science-fiction fans talk fast about mathematics. They also do a lot of eccentric computer programming.

Many of these people believe their goal is to create intelligent (and possibly conscious) entities: new beings that will populate the world of the future — either living amongst humanity or replacing it (the radical view). This is not the only goal.

The patrons, for instance, have concerns of their own. The patronizing patronage of the Defense Department and the Intelligence Community is based on eagerness for the real or fancied products of these endeavors — automatic translation of what is written or spoken in Enemy or Third World lands; ways of automatically scanning satellite photographs for missiles, tanks, etc.; things a cruise missile can scan the radar or TV for on its merry way to a target planned in advance.

But there are many other goals. For instance, radical changes in education. The AI freaks tend to be brilliant people from well-to-do homes who hated school, in part because they were much smarter than their teachers, and who hold an idealized view of how wonderful education could be if only we could free children from its more oppressive aspects.

A few maverick AI people are obsessed with comparing their ideas to empirical reality. Now this minority is breaking away to form a coalition with psychologists under the name "cognitive science." But sections of the field have broken off before, and many others will again. Artificial Intelligence is the great glacier from which so many other icebergs have broken off; the remainder will always be immense.

CONTINUED ON PG. 90

An Actor-Based Animation Language

Kenneth M. Kahn

Originally circulated as AI Working Paper #120 and Logo Working Paper #48. This work was supported in part by the National Science Foundation under grant number GJ-1049 and conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program. Reproduction of this document in whole or in part is permitted for any purpose of the United States Government.

Section I. Introduction

Some of the recent AI languages are based on a new view of computation sometimes called "actor" semantics. Carl Hewitt's PLASMA ([Hewitt 1975] and [Smith 1975]) and Alan Kay's SmallTalk [Goldberg 1974] are the best examples. The basic idea is to consider each entity within the system as something that is usually anthropomorphized as a "little person." Each "little person" or "actor" can receive messages asking it to do something, remember something, recall something, or send some messages to other actors. For animation this seems an ideal way to represent objects on the display. Each object is a process that can be arbitrarily smart. Charlie Brown can be an "actor" that can be told to walk, causing him to send the appropriate messages to his arms and legs and moving the rest of his body.

Ken Kahn, Computer Science Department, University of Stockholm, Stockholm, Sweden.

CONTINUED ON PG. 76

Programming And Animating On The Same Screen At The Same Time

Paul Pangora

I think there is much more work to be done in the area of matching the programmer's approach to the software environment, and I spread this philosophy whenever I can. My focus in this article is the importance of matching the concept of a system to a sensible programming structure, for ease of translation from idea to executable software.

Paul Pangaro, 212 East 48th, New York, NY 10017.

CONTINUED ON PG. 86

Erratum

If you are saving our October issue, please change the word "Smalltalk" on page 66, column 2, paragraph 4, line 1 to "Sketchpad." You probably understood it anyway.

We asked Adele Goldberg who should be credited with Smalltalk, and here is what we got back:

Smalltalk development:

We would like to credit the members of the Xerox Palo Alto Research Centers on whom the ideas and their realizations depend, but this is a much shorter list than "everyone" due to space limitations.

This is a list of the *current members* of the Learning Research Group, Systems Science Laboratory, Xerox Palo Alto Research Centers, in alphabetical order:

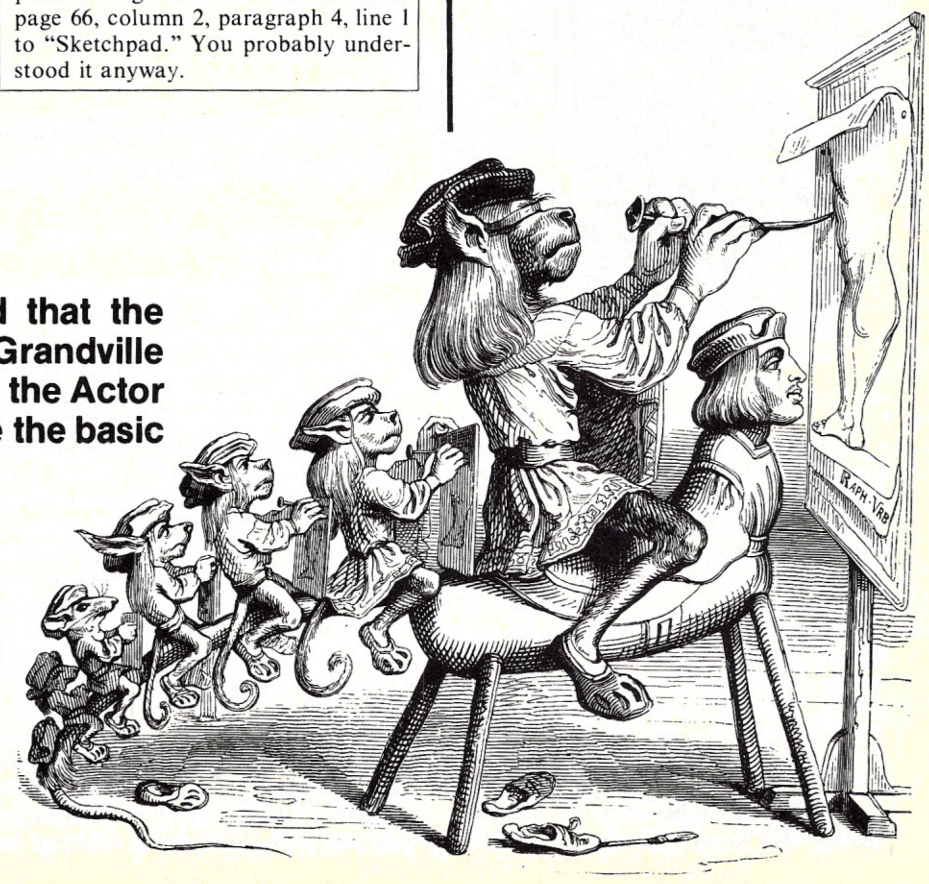
Jim Althoff, Al Borning, Peter Deutsch, Robert Flegal, Adele Goldberg, Laura Gould, Bruce Horn, Dan Ingalls, Ted Kaehler, Alan Kay, Glen Krasner, Kim McCall, Diana Merry, Janet Moreland, Steve Putz, Dave Robson, Steve Weyer.

Past Heroes: *Karla Garcia, Chris Jeffers, John Shoch, Larry Tesler.*

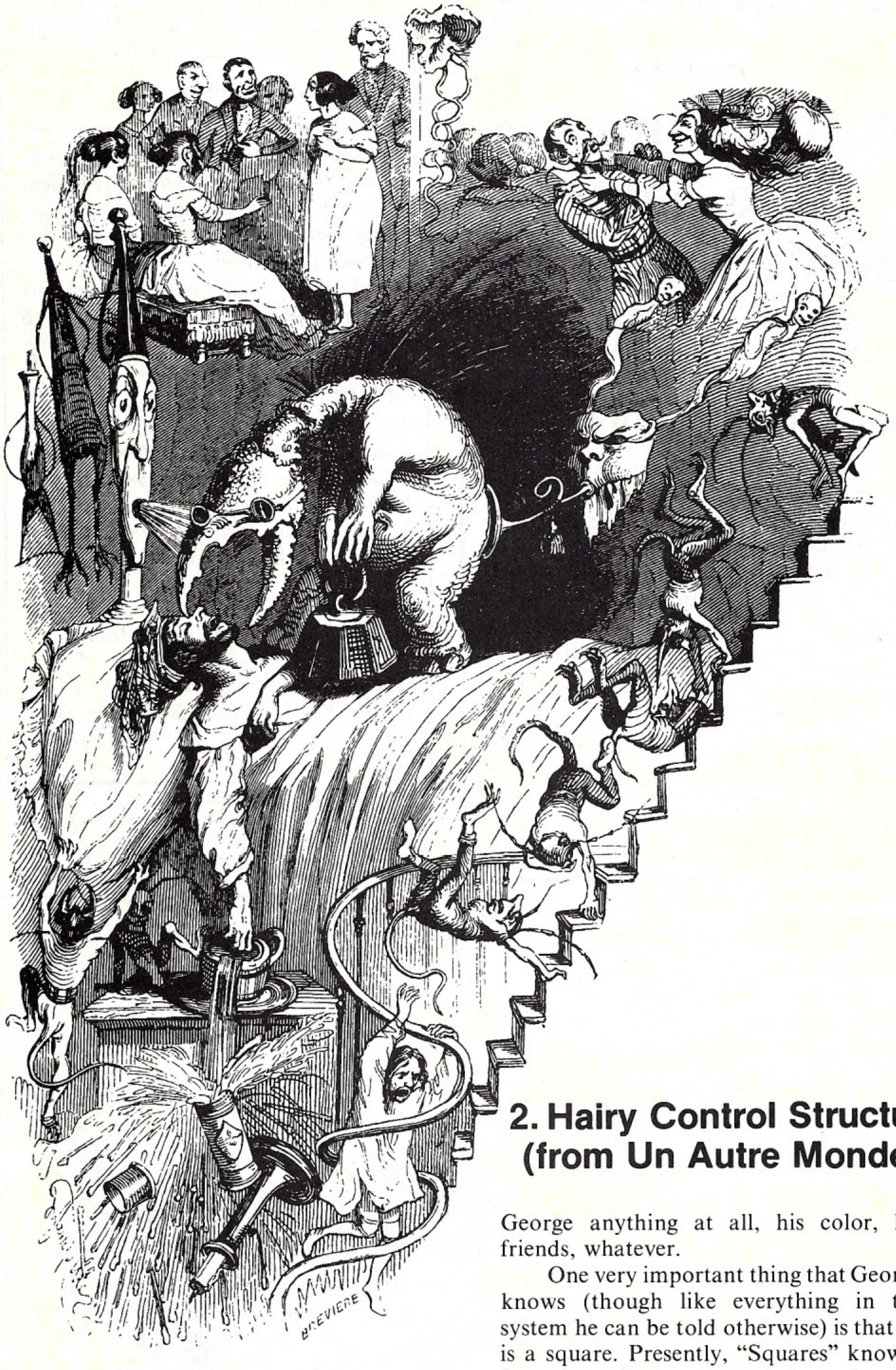
Students: *Barbara Koalkin, Vicky Parish, Steve Purcell, Bob Shur, Dave Smith, Todd Snook, Jim Stamos, Fred Tou.*

"Three Smalltalks" by Mark Miller, cited last month as appearing in this issue, has been withdrawn at the request of interested parties. We hope to be able to publish it at a later time.

It is not generally realized that the great French cartoonist Grandville (1803-1847) clearly foresaw the Actor languages. We present here the basic concepts as he saw them.



1. Recursive Graphics (from La Vie des Animaux)



2. Hairy Control Structures (from Un Autre Monde)

I have implemented a system in Lisp-Logo [Goldstein 1975] which enables one to define new objects, new object types, and the kinds of messages they can handle. For example, one can easily create a square named "George." George can be told many kinds of things like his size, speed of movement, or speed or rotation. George can be asked to do many things, all the things that turtles can do (FORWARD, BACK, RIGHT, LEFT, HIDE, SHOW, PENUP, etc.) plus new acts like growing, or changing appearance. George can also be taught new things, or can be told to behave in ways other than his defaults. George also has a memory; you can tell George anything at all, his color, his friends, whatever.

One very important thing that George knows (though like everything in the system he can be told otherwise) is that he is a square. Presently, "Squares" know a few things, like how to draw themselves, or that after rotating 90 degrees that they look the same. Squares in turn know that they are instances of "Object." Objects know how to do the turtle-like things mentioned above. Objects, in turn, know that they are instances of "Something," things that can receive messages, can pattern match those messages, and can perform memory functions. "Somethings" know how to learn new responses to new patterns. This entire hierarchy is very flexible and modifiable by the user. The basic process is the message is sent to some individual, "George," and if George has no patterns that match the message, he sends it off to the actor that he is a kind of. They in turn pass the buck, until either someone can handle the message, or an error message is generated. This is also a very useful default mechanism; if George is never told his size he can inherit it from "Square" or "Object."

Another feature of this system is the ability to have many different actors move on the screen with apparent parallelism. George can race against Sally. Danny's garden of flowers can be grown [Hillis 1976]. A stick figure can simultaneously move different limbs and change its facial expression. "Movies" (or a list of display commands) can be produced that can be run forward or backwards at any speed the computer system is capable of or single stepped.

This system is intended to support the intelligent computer animator which is discussed in the companion AI Working Paper 119 (Logo 47) and, equally important, to be used by children. The system is hopefully a more powerful and natural means for doing simple programs for animation. The powerful ideas of "instances, classes and finding of the correct level of generality" and the "little person model of computation" are imbedded into the system. The hope is that through well-guided use of the system, some of these ideas will become more concrete to the children. Of course, all the usual reasons for teaching Logo to children remain in force (e.g., learning by doing, experience with debugging, becoming articulate in describing processes, and exposure to and assimilation of powerful ideas). The use of the same actor-animation system by children and by the computer animator is very important for making the computer animator more accessible and understandable by the children using it. The idea is that if the children who programmed using the actor-animation system found it natural and intuitive then its use in the intelligent system would also be clear.

This view of programming as collections of actors, or a community of "little people," that send and receive messages from each other is very powerful. It is conducive to a modular, simple, natural representation of the knowledge needed for the application. Using an actor system one can model intelligence as an integrated community of rather limited individuals or in the more conventional manner as an integrated individual.

Another AI aspect of this system is the explicit "kind-of" hierarchy of actors. Each object is told what class it is a member of when it is created. When any object receives a message it cannot handle it passes the problem on to the class of which it is a member. The important concepts of instantiation, class membership, exceptions, placement of knowledge at the best level of generality, and inheritance of properties hopefully will flow from the proper use of this aspect of the system.

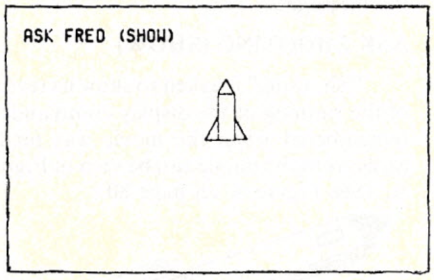
To both facilitate the use of the system and to give the user a good start in what is hopefully the right direction, the top nodes of the hierarchy can be predefined. Currently the default hierarchy consists of an actor called "Something" at the top node. "Something" can receive messages to remember, forget, replace or find items. It receives messages for editing actors including ones to insert, remove, replace, interchange or correct receivers. It also can create new instances of actors, print out the script and memory of an actor, and accept messages to be done at a later time. "Object" is an actor directly below "Something" and as such inherits all of its abilities.

In addition "Object" can behave like a Logo turtle on the tv display. "Objects" also can move across the screen at a particular speed, can rotate at any speed, can revolve around a point, move away from a point, grow and shrink. "Object" also remembers various items such as its rotational speed, size, and speed thereby providing default values to all its inferiors. There are other sub nodes of "Something" such as "Movie" and "Universe" which are discussed later. Under development are "Composite-something" and "Composite-object" which know how to send appropriate messages to their parts. In the following examples "Rocket" and "Flower" which are instances of "Object" are used for illustrative purposes.

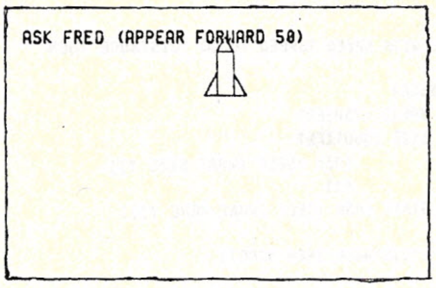
Section II. An Example

ASK ROCKET (MAKE FRED)

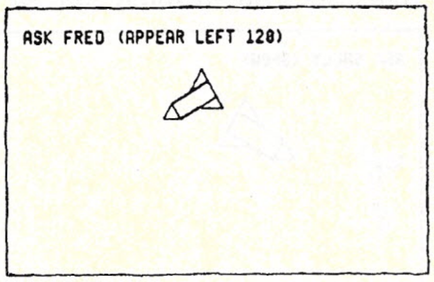
ASK is the basic message passing command, in this case the message "Make Fred" is being sent to the actor "Rocket." "Rocket" does not know how to handle messages of this form so it passes the message to "Object" who is also ignorant of such messages. The message is finally sent to "Something" which can match the message with one of its patterns and it creates a new actor named "Fred" which is a kind of "Rocket."



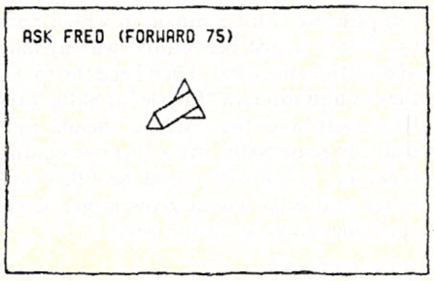
Fred is asked to show himself. He knows nothing about "showing" and asks "Rocket" who asks "Object" which can handle the message. It asks Fred what turtle procedure draws him and Fred doesn't know so asks Rocket who answers with the name of a standard Lisp (or Logo) procedure. It then asks Fred for his position, heading and size and then invokes the Lisp (or Logo) procedure.



"Object" is passed this message via "Fred" and "Rocket" and Fred is asked to Hide and then to appear at the place 50 steps forward. The word "APPEAR" is there to distinguish this type of message from those in which the movement is gradual as described later.



Again "Object" handles this type of message and Fred is asked to Hide, then to rotate to the left 120 degrees and then to show.



This time Fred is told to go forward, so he asked for his speed; he has none in this example and asks "Rocket" for his speed, which is 25. He then is asked to (APPEAR FORWARD 25) and to plan on continuing the rest (50) on the next tick of the "clock."

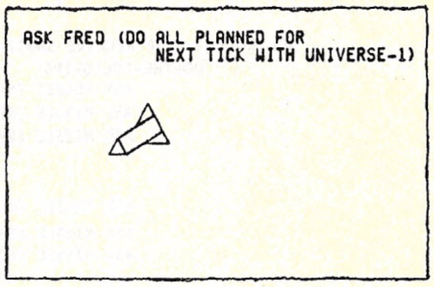
ASK FRED (WHAT SPEED ?)

25

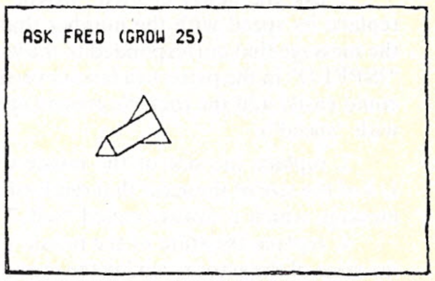
In responding to the previous message Fred was asked what his speed was. Fred, as is true of all "Somethings," has a memory. This memory is a general relational data base and it is also used to maintain the state of actors. The message "What . . . " indicates that the value found to is to be returned.

ASK FRED (REPLACE SPEED 50)

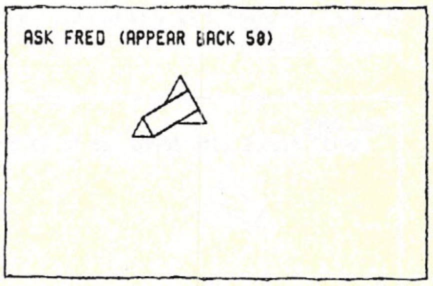
Fred is asked to remove the item about his speed (there is none in this case) from his memory and to remember the item "(SPEED 50)."



Fred is asked to do all the things that he had planned to do on the next "clock" tick.[1] He asks himself what things he had planned then and does them. In this case the only thing that was planned was "(FORWARD 50)" which was left over from earlier. Since his speed is now 50 he can do it all and appears forward 50 steps. "Universe-1" is an actor that can be asked for all the other actors that are currently on the screen, so that interactions are possible. For example, collision or avoidance can be implemented by asking the other actors where they are and maybe even where they are planning on going.



Fred is told to grow, which causes a message to be sent to him to hide, then to replace his size with his old size plus 25 and finally to reappear again.



Now when Fred is told to do anything an image of him with his new size moves.

1) To some people this wordy style of programming is distasteful. I could just as well have defined the message to be "(TICK UNIVERSE-1)". It is very important, however, that the code be as clear and easy to read as possible. The difficulty in typing can be overcome by simple human engineering aids; for example, a special "help" button which, when pushed, could finish the line to the extent possible, saving much typing and preventing misspellings.

```
ASK ROCKET '(IF RECEIVE SHOOT MISSLE WITH SPEED ?SPEED TO GO ?DISTANCE THEN
          DO-THE-FOLLOWING:
                ASK ROCKET (MAKE MISSLE)
                ASK MISSLE (REPLACE SPEED :?SPEED)
                ASK MISSLE (REPLACE SIZE (QUOTIENT
                                          (ASK :SELF (WHAT SIZE ?))
                                          4))
                ASK MISSLE (REPLACE STATE (ASK :SELF (WHAT HERE ?)))
                ASK MISSLE (SHOW)
                ASK MISSLE (FORWARD :?DISTANCE THEN HIDE))
```

The behavior of any actor in the system can be extended. The "if receive..." message is matched by "Something" which adds a new receiver to the actor that received the message. In this case, "Rocket" is sent the message asking it that if it receives any messages of the form: the word "shoot" followed by the words "missle with speed, " then any word, then the words "to go" followed by only one more word,[2] then call the first word "?speed" and the second word "?distance."[3] Then do the following series of things:

1. create a rocket named "Missle" (it is possible to make the name "Missle" local to this receiver or to have a unique name generated)

2. ask the newly created "Missle" to replace its speed with the number that in the message that corresponded to the word "?SPEED" in the pattern (a fancier version could easily add the rocket's present speed with "speed")

3. replace the size of the missle with 1/4 of the size of the actor that received the message which is always called "self."

4. replace the state of the missle with the state of the actor receiving the message; this way the missle appears where the shooter is, rather than the default which is the center of the screen.

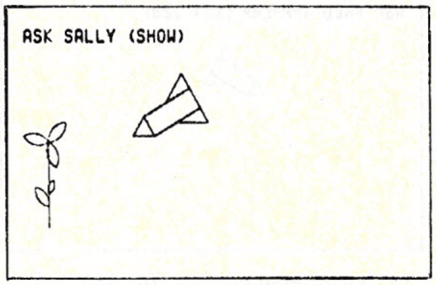
5. the missle is asked to show itself.

6. it is told to go forward the last word in the message.

7. finally, it is told to hide when the finished moving forward (a fancier version might explode).

Fred is given this newly defined type of message and then asks "Rocket" to try to handle it. It can, and the above procedure is executed with the speed of the missle being 100 and the distance it is to travel being 200.

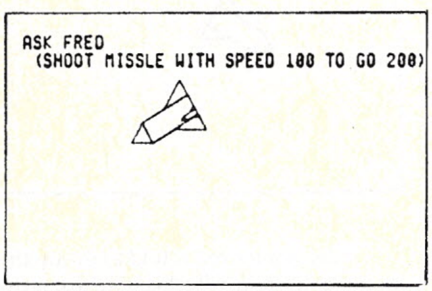
ASK FLOWER (MAKE SALLY)
ASK SALLY (REMEMBER SIZE 50)

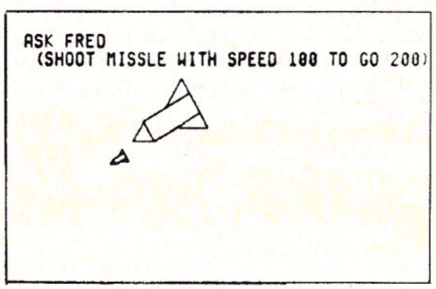


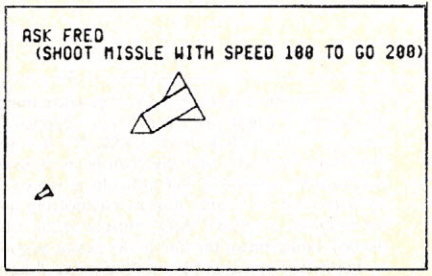
A flower named Sally is created, given a size and asked to show.

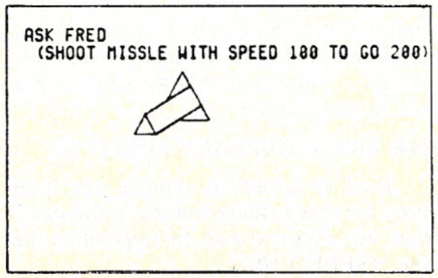
Now we can use "Flower" and "Rocket" to make a little animated movie. Suppose we want a movie in which Sally the flower is just peacefully swaying back and forth in the wind. Then Fred the rocket flies by and shoots a "Missle" at Sally. Fred flies away and the "Missle" heads right towards poor Sally. As a surprise ending, however, the missle could be filled with water and Sally could grow larger as she continues to sway in the wind.

ASK SALLY
(PLAN: SWAY 10 DEGREES
12 TIMES NEXT)









"Flowers" can be asked to accept "sway" messages which cause them to go left and then right the specified number of degrees. The "Plan:" part is the same kind of message that "Forward" produced previously. Sally does nothing on receiving this message other than remember to do it with the next tick of the "clock."

ASK FRED
(PLAN: FORWARD 300 NEXT)
ASK FRED
(PLAN: SHOOT MISSLE WITH SPEED 50 TO GO 150 NEXT)
ASK FRED
(PLAN: RIGHT 90 THEN SHRINK 100 AFTER 2 MORE TICKS)
ASK SALLY
(PLAN: GROW 60 AFTER
6 MORE TICKS)

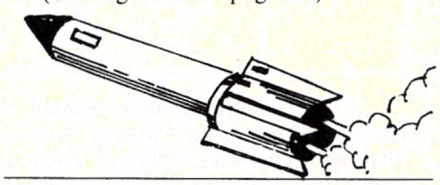
More events are scheduled, such as Fred being told to begin going forward 300 steps and then shoot a missle with a speed of 50 to go 150 steps. Two "clock" ticks later Fred will start to turn right 90 and when finished turning will shrink away. Notice that he will be turning while he still has some steps to go forward and thus will plot a polygonal course. Six frames into the movie Sally will begin to grow. (The numbers of ticks in this example were chosen simply because they caused things to happen at the right time. Much of this would become simpler if one could plan events relative to other events. This is a problem I hope to tackle soon.)

ASK MOVIE (MAKE SHOOTING
12 TICKS LONG IN UNIVERSE-1)

Here "Movie" is asked to make a movie called "Shooting" that is 12 frames long. It in turn asks "Universe-1" to send to a "tick" message to all the actors with things planned. It will stop either after 12 ticks or sooner if no more things are planned. The "Screen" asks "Shooting" to remember each display command in addition to doing them, so that they can be played back at a speed that is not limited by the time it takes to send and interpret all the messages.

ASK SHOOTING (SHOW)

"Shooting" is asked to show its record of the running of the display commands it remembered when the movie was made. Stills from the movie can be seen in Figure A. (See Figure A on page 80.)

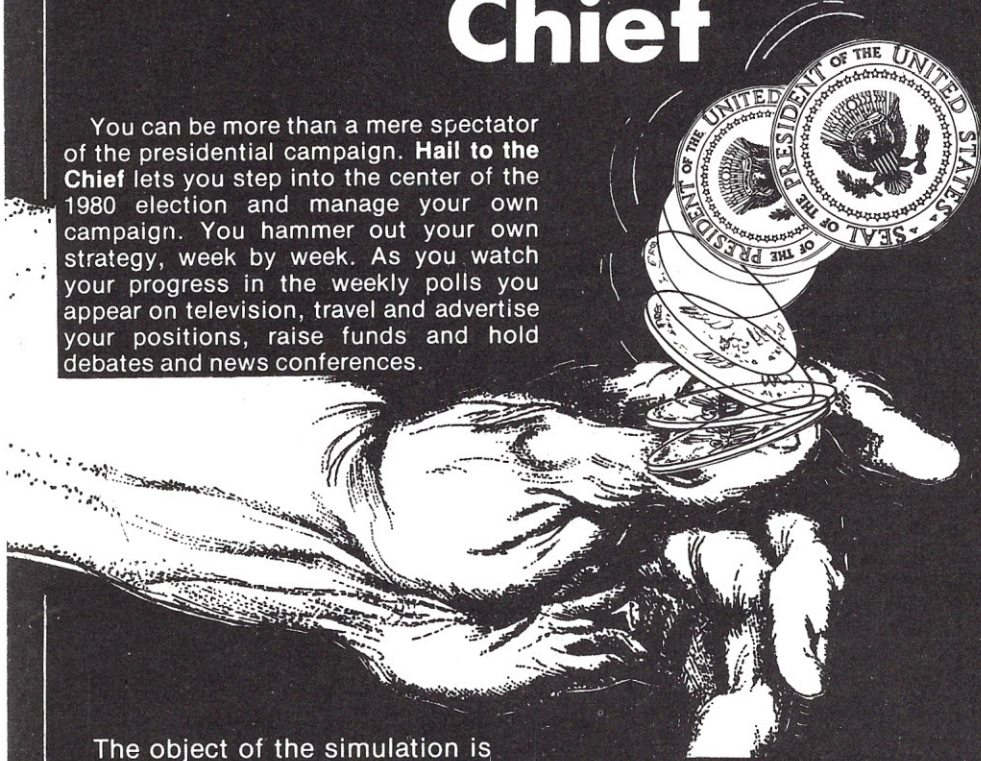


2) If the pattern was SHOOT MISSLE WITH NUMBERP ?SPEED TO GO NUMBERP ?DISTANCE then it will match only if the words following "SHOOT" and "GO" are numbers.

3) The ":" in front of the names is a convention necessary to be compatible with Lisp-Logo. It is also used in messages to indicate that the value of the atom, not the atom is intended.

Hail to the Chief

You can be more than a mere spectator of the presidential campaign. **Hail to the Chief** lets you step into the center of the 1980 election and manage your own campaign. You hammer out your own strategy, week by week. As you watch your progress in the weekly polls you appear on television, travel and advertise your positions, raise funds and hold debates and news conferences.



The object of the simulation is to become president-elect when the final election results are in. Refinements of the simulation such as the influences of incumbency, campaign finance and spending limits are introduced in increasingly complex models. Each model can be played at ten levels of difficulty—a level 10 opponent is tough to beat.

This is a straightfoward simulation, without scenarios of blatent corruption, but temptations to compromise your ideals are still realistic and powerful...

Will you change your positions to capture the financial support of labor of Big Business? Or play down your unpopular positions to capture the votes of a particularly important and sensitive region? As in real life, the presidency can be captured by integrity or guile. Only the candidate knows the price of his, or her, success.

The ten pivotal camgaign issues in **Hail to the Chief** include Energy Policy, Unemployment, Mid-East Policy and Strategic Arms Limitations. Your positions are chosen from a nearly two hundred degree numerical scale which ranges from "Bleeding Heart Liberal" to "Middle-of-the-Road" to "Reactionary". For example, strong conservative and liberal statements on Strategic Arms Limitations are:

Our enemies understand only strength and they have proven that they will not honor any treaty obligation that stands in their path toward world domination.

An uncontrolled arms race is more likely to lead to war then any other policy the U.S. can follow; it is also a waste of our resources and puts too much power in the hands of the military industrial complex.

Hail to the Chief has been used as a teaching aid in Political Science, Computer Science and Voting Behaivior courses at the University level since 1976. Its authors are Associate Professors at the Eastern Kentucky University; Phillip W. Brashaer in Mathematics and Richard G. Vance in Political Science. A comprehensive manual, discussion questions and background materials have been prepared by the authors and accompany the fun and educational package. **Hail to the Chief** is available for the TRS-80 level II on a 32K cassette (CS-3205) and a 48K disk (CS-3701), for the Apple II and Apple II Plus on a 48K disk (CS-4704), for the Atari 400 and 800 on a 32K cassette (CS-7201) and for the Atari 800 on a 40K disk (CS-7701). All are $24.95.

Your local computer store should carry Creative Computing Software. If your favorite retailer does not carry the software you need, have him call in your order to 800-631-8112. Or, you can order directly from Creative Computing. Creative Computing Software, Dep't. AHGG, P.O. Box 789-M, Morristown, NJ 07960. Include $1.00 for postage and handling. For faster service, call in your bank card order toll free to 800-631-8112.

sensational software

CIRCLE 300 ON READER SERVICE CARD

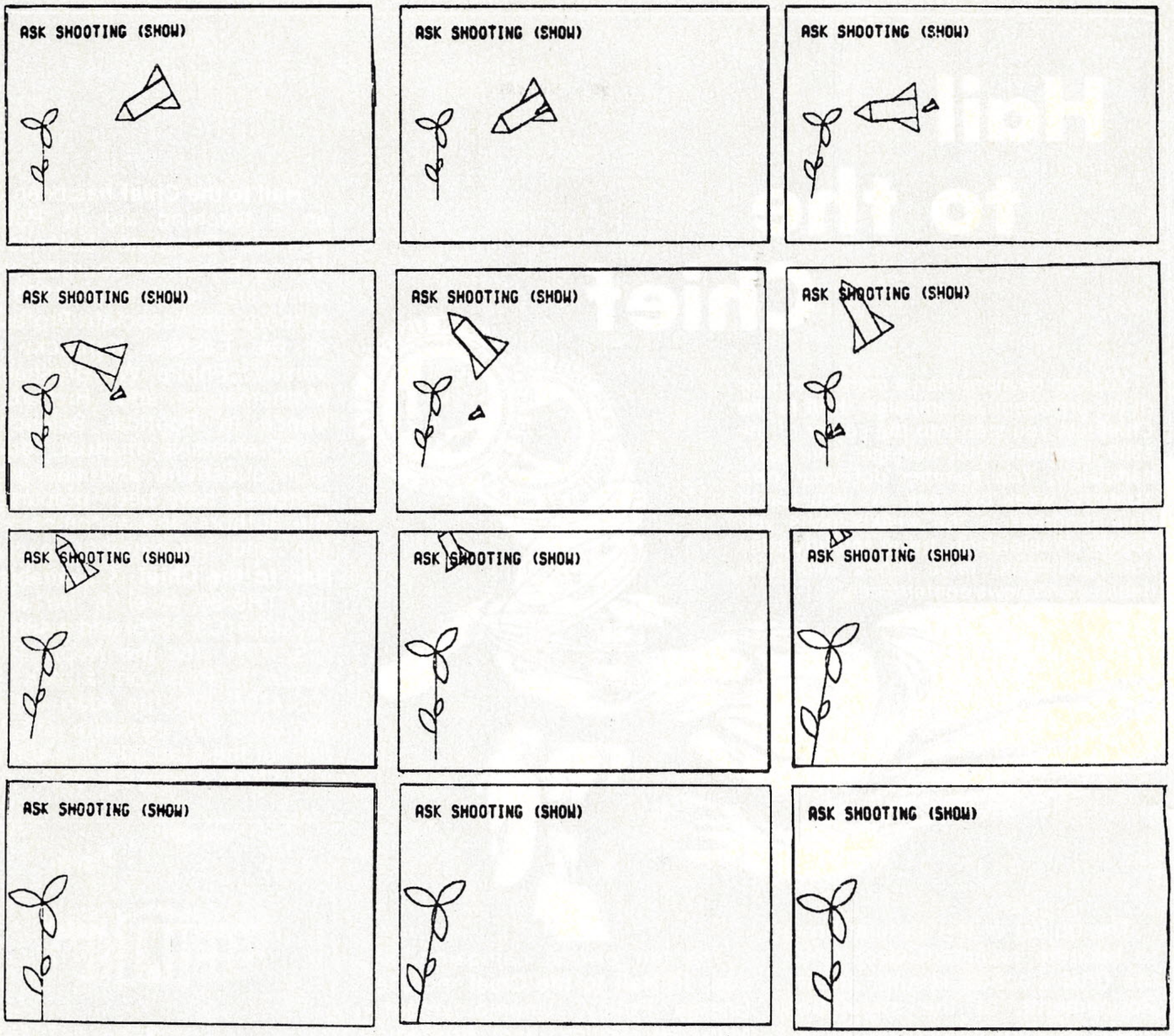


Figure A.
Scenes from the "Shooting" Movie.

Section III.
How to Grow a Flower Garden

In a paper by Danny Hillis called, "Ten Things to do with a Better Computer," he has an example of how to grow a garden in an actor-like system [Hillis 1976]. He describes a garden in which seeds are born, wait, grow into flowers, create new seeds, continue growing and die.

In my animation system, his garden can be implemented in a fairly straightforward manner as follows:

TO DEFINE.FLOWER

Logo syntax can be used if my system is loaded into Lisp-Logo

10 ASK OBJECT (MAKE FLOWER)
20 ASK FLOWER (REMEMBER SIZE 10)

This tells flower to remember that the default size of flower is 10. "Remember" is the standard kind of message for telling any actor to remember something

30 ASK FLOWER (REMEMBER DRAW USING DRAW-FLOWER)

This tells flower that the Logo procedure called "draw-flower" is to be used to draw instances of "Flower."

END

TO DEFINE.SEED
10 ASK SOMETHING (MAKE SEED)

Seed is not an Object, since it does not do turtle-like things.

20 ASK SEED (IF RECEIVE ?SEED (START) THEN DO.SEED.THING :?SEED)

This simply lets seeds take a "start" message and then calls the appropriate procedure.

END

TO DO.SEED.THING :SEED
10 LOCAL A.FLOWER

A local name for the flower that seed will spawn is needed.

20 ASK FLOWER (MAKE A.FLOWER)
30 ASK A.FLOWER (APPEAR RIGHT 90)

Flowers are "Objects" and so can take any turtle-like command.

40 ASK A.FLOWER (APPEAR FORWARD (* 100 (RANDOM)))
50 ASK A.FLOWER (APPEAR LEFT 90)
60 ASK A.FLOWER (PLAN: SHOW IN 10 TICKS)

The message transmission ASK A.FLOWER (SHOW) will occur after "A.flower" has received 10 "Tick" messages.

70 REPEAT 15 (ASK A.FLOWER (PLAN: GROW 10 AFTER 2 MORE TICKS)

This schedules the call, ASK A.FLOWER (GROW 10)
15 times, each time 2 ticks after the last.

80 ASK SEED (PLAN: ASK (SEED (MAKE)) (START) AT THAT TIME)

This means at that time create another seed and tell it to start at the same time as the last thing scheduled, which was after line 70 was run.

90 ASK A.FLOWER (PLAN: HIDE AFTER 60 MORE TICKS)
END

THE PROGRAM STORE (202) 337-4691

4200 Wisconsin Ave.NW P.O.Box 9609 Washington D.C. 20016

Osborne & Associates ACCOUNTING SYSTEMS

For two years we have looked for good business accounting programs. After reviewing many over-priced or non-functioning packages, we have finally found some excellent software. And it is available at a reasonable price - $25.00 per package.

These programs are the Osborne & Associates systems which are copyrighted by Taranto & Associates. They are excellent programs which have been used for several years. Originally written for other computer systems, these programs are now available for the 48k Model I TRS-80 with two disk drives.

These programs are not games. They are meant to adapt to your accounting system. You should not expect us or this software to be your accountant. But, if properly used, you will save hours of valuable business time.

All Osborne & Associates programs are fully documented by their three extensive books - "General Ledger," "Accounts Payable & Accounts Receivable," and "Payroll with Cost Accounting." (These TRS-80 programs do not include the cost accounting system. The General Ledger does contain a Cash Journal.) These books are available for $20 each. Please include $3.00 per book for first class shipping (otherwise sent U.P.S.).

Act today!

General Ledger	$25.00
Accounts Receivable	$25.00
Accounts Payable	$25.00
Payroll	$25.00
Any book	$20.00

SUPER NOVA

by Bill Hague from Big Five

Asteroids* surround your ship. Shoot the asteroids and the alien ships. Written in machine language, this game is GREAT! There are five different types of alien ships including the very deadly Flagship.* You shoot from your ship, rotate it, use your thrust key to move, and in emergencies you go into hyperspace. Level I or 2 - $14.95 *

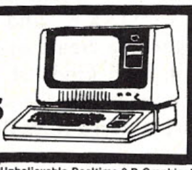
GALAXY INVASION

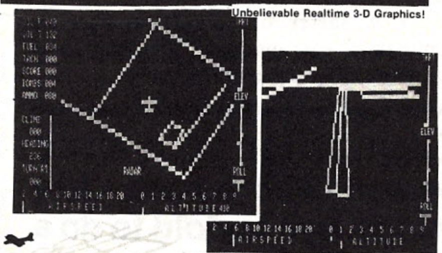
by Bill Hague from Big Five

"Galaxian" is the rage at the arcades. Now GALAXY INVASION is the rage on the TRS-80. The aliens move about the top of the screen and will smoothly swoop down on you. But watch out for the Flagship Alerts! Level I or 2 - $14.95

(P.S. This ad almost did not get finished due to Galaxy Invasion addiction.)

TRS-80 16k Level II Programs





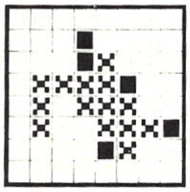
FLIGHT SIMULATION

by Bruce Artwick from Sublogic

The wait is over! If 3-D graphics seem impossible on the low resolution TRS-80, you haven't seen this brilliant program. During FLIGHT SIMULATION, you instantly select instrument flight, radar, or a breathtaking pilot's-eye-view. But be sure to strap yourself in -- you're liable to get dizzy!

Once you put in some air time learning to fly your TRS-80, head for enemy territory and try to bomb the fuel depot and airstrip while fighting off five enemy warplanes. Good luck!

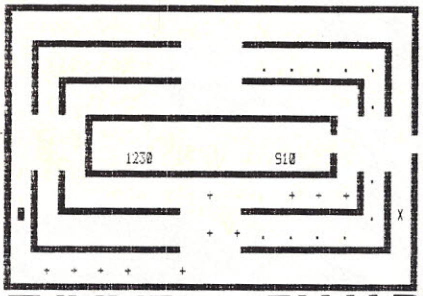
Level I or II protected cassette $25.00



OTHELLO

from Instant

We have long looked for a good Othello playing program. This is it! The program has seven levels of play and it will let you see its thinking. Written in machine language, you decide where to move using the arrow keys. $9.95



TUNNEL OF FAHAD

by K. Pfeiffer from Adventure

Patterned after the popular arcade game "Crash", this is an action game with sound. You attempt to recover the silver and tanner leaves while avoiding the Mummy. $9.95

QUAD

by Charles Asper from Acorn

If you are looking for a logical and challenging game you should try your hand at QUAD. The game is like 3D tic-tac-toe with a time clock and four levels of play for one or two players. Vivid graphics and six-way cube rotation so you can see the play from any angle.

Protected tape $14.95

Protected disk $20.95

BOSS

by V. Hester from Soft Sector

This utility is the perfect tool for creating and debugging BASIC programs. It allows single stepping through a BASIC program, setting up to five break points in the program, and tracing of program logic using only a small portion of the screen. You can also review selected variables during program execution and return to your program with the display restored. If you like, you can store one or more programs in high memory for later retrieval.

Known to work with Level 2, TRSDOS, NEWDOS 2.1, NEWDOS/80, and VTOS. $29.95

THE BØØK

ACCESSING THE TRS-80* ROM

from Insiders Software Consultants

For the machine language expert or beginner! This first of three volumes gives you access to over fifty arithmetic, mathematical, and data conversion routines in the TRS-80 ROM. Includes sample programs, a disassembler program, a commented listing of these ROM routines, and a memory map of over 500 addresses. 134 pages. $14.95

DISK*MOD

by Roy Soltoff from Misosys

This machine language program modifies your copy of the Radio Shack Editor/Assembler for use with your minidisk and any disk operating system. You can save and load both text source and assembled object files. Unlike the NEWDOS version you can read the directory and the allocation of granuales while in the EDTASM. You can also kill files. It is a complete disk modification for one or more drives.

Other capabilities are also added which are not found on NEWDOS. The block move command relocates a section of text to any other area. The global change command permits, for example, changing a label throughout the text. The pagination feature provides hardcopy on 8 1/2 by 11 pages on either single sheets or continous paper. In addition, high memory can be reserved, like in BASIC, for machine language routines like printer drivers. You can also display the amount of memory remaining.

The [CLEAR] key is functional, the symbol table is sorted alphanumerically and output 5-across, the scroll up/down allows 15 lines on the screen, and the 'DEFM' assembly is improved. Lower case input is now permitted and you can branch to any address. Plus, it also corrects the errors in the Radio Shack tape version.

Save your time and make full use of your disk system by upgrading your Editor Assembler today. $19.95

THE PROGRAM STORE
4200 Wisconsin Ave NW
P.O. Box 9609 Dept. R5
Washington DC 20016

VISA AND MASTERCARD

TO ORDER TOLL FREE 800-424-2738

For program information call (202) 337-4691

Mail orders: Send check or money order for total purchase price, plus $1.00 for shipping & handling. Charge card customers, please include card number, expiration date, and your name as it appears on the card.

CIRCLE 175 ON READER SERVICE CARD

Actor-Based, cont'd...

ASK (ASK SEED (MAKE)) (START)

This starts the first seed off which will start the others off later.

ASK MOVIE (MAKE GARDEN 1000)

The default universe sends ticks to all the flowers and seed involved, the things they had planned are run, and all display and turtle commands are stored in the movie "Garden."

ASK GARDEN (REMEMBER SPEED 10)

ASK GARDEN (SHOW)

These two message transmissions result in the movie being shown at 10 frames per second. First, a flower appears and grows and then another starts to grow, then another and after a while the first flower disappears and so on.

Section IV. Message Passing

Message passing is the mechanism of communication between two actors. A pattern matching procedure called "Match" is used to decode the incoming message. This provides great flexibility in the syntax of the messages sent and received, and enables the use of much more readable commands.

The following is a list of some of the messages that any instance of "Something" can receive:

1. The "RECEIVE" message which enables an actor to increase the set of messages it understands.
2. The memory messages, "REMEMBER," "WHAT," "REPLACE," and "FORGET" which provide a powerful relational memory to each actor. These are also used to inspect the position, heading, speed, size and the like of an object and to update those quantities.
3. "MAKE" and "UNMAKE" messages which create or destroy instances.
4. Two kinds of "PRINT" messages which print out the script or the memory of an actor in a form designed to be easy to read.
5. A series of structural editing commands for inserting, deleting, and replacing receivers or their parts.

Each actor is a little person, who knows a few things, can be told to do a few things, and can be taught new tricks.

6. "PLAN:" message for scheduling things to do at later times.
7. The "TICK" message for telling an actor to do all that he or she should during the smallest quantum of time and providing him or her with access to the names of other actors, enabling interactions.

Instances of "Object" can, by passing the buck to "Something," receive all the above messages. They can also receive the "turtle" commands, i.e., "FORWARD," "BACK," "RIGHT," "LEFT," "HIDE," "SHOW," "PENUP," "PENDOWN," of which there are two varieties. One, the object moves gradually across the screen at some speed while the other the object disappears and appears at the new position.

Section V. Ticks Plans and Movies

Using the system, one can write procedures that move one object, then another and back again. It will not, however, look as if they are moving simultaneously since the interpretation and execution of the commands is slower than the maximum duration for persistence of vision of occur. A solution is to have a scheduler run things at the appropriate time, and then save away the display commands to be run later. These saved-away commands, called a "Movie," can be run later with the appearance of parallelism. This was done in earlier implementations but the current one distributes the responsibility to each actor to remember what it will do later. The scheduler, or "Universe" as it is called, just sends "tick" or "increment the time" messages to all actors it knows about. Movies can be remembered or just the code run, depending on the message to the "Universe" or "Movie."

If one wants an event to be dependent upon the occurrence of another event then the appropriate actors must check for the occurrence of the event when it receives a tick message. For example, if Lucy is told to scream if Snoopy comes too close then the "Lucy" actor must check to see where Snoopy is whenever she receives a tick message. If Lucy was told to scream if anybody came near, she would need to know the names of everybody around. That is why name of the current universe comes along with each tick message. Lucy can ask each actor where he or she is.

When an actor is told to plan something it is always relative to that own actor's internal time. His or her time is incremented with each tick. Associated with each time are the things that actor plans to do at that time. After the actor remembers the things planned it tells the current universe that it has things to do and would like to be placed on the mailing list for receiving tick messages.

Message passing is the mechanism of communication between two actors.

The universe when told to run will send ticks to each actor with things to do. When an actor has nothing to do it tells the universe who stops sending it ticks. When the universe has run the number of ticks asked of it or there are no more actors with things to do it stops.

The screen is an actor that receives display messages like "Put George Forward 100." When a movie is being made the screen can send messages asking the movie to remember the display messages the screen received. The movie can then be run later without sending any messages except those to the display.

Actor-Based, cont'd...

Section VI.
How Does One Talk About This?

In teaching children to use the animation system one should have a consistent vocabulary and set of concepts. Are the words "Something," "Universe," and "Object" reasonable names for these general entities? I think not, but haven't thought up better ones. How should one talk about the difference between "Square," the general square, and any instantiation of squares? Will there often be a confusion between the actor "George," his "script," and his image on the screen? Identity becomes very strange to talk about, since anything about an actor can be changed; what it looks like, how it acts, even what it is a kind of. (Though this programming style may be discouraged to avoid this confusion.) Only its name is permanent.

Explaining the ticks, plans and universes might bring up problems. How should one talk about, or think about, many separate processes going on in parallel? I don't really know.

This system lends itself very well to the "little man" vocabulary. Each actor is a little person, who knows a few things, can be told to do a few things, and can be taught new tricks. This little person receives messages and sends them to others. This view of programming lends itself very well to avariation of playing "turtle." The children can play games where they are actors and pass the messages around. This link between reality and this programming style I hope will make the system seem more intuitive and natural.

Section VII. Efficiency Issues

One may worry that such a system will run too slowly to be useful for working with children or for building an intelligent system on top of it. The message passing and matching involved are much slower than more traditional mechanisms. The basic use of hierarchies is slow, since each actor seldom can respond directly to a message but needs to pass it on to the class which it is a member.

The answer to this objection is standard. One should let a compiler worry about such efficiencies. I have implemented a few macros that when possible replace actor transmissions with the code that they would invoke. I also plan to "compile" patterns in the receivers to run faster. The price for some of these hacks is less flexibility. If the transmission "ASK FRED (APPEAR FORWARD 100)" is replaced by the action part of the receiver in "Object" with "Fred" and the amount instantiated properly. However, telling Fred's immediate superior a new way to handle "forward" messages will not affect his behavior if he is "compiled."

There are other efficiency hacks that may be worthwhile, for example, in the memory system for the actors. I plan, however, to follow the principle that the code should be written clearly and simply and that efficiency hacks should be below the surface and transparent to the user.

Section VIII.
Extensions and Improvements Planned

One useful extension would be the addition of primitives for joining and breaking apart "Object" actors. For example, one may want to join a triangle actor and a square actor to make a house actor. Or one may want to have a face accept messages as well as any of its parts. A person may get into a car, so that temporarily any movement of the car should also change the state of the person. Some progress has been made here, so that simple composite objects can be defined but more needs to be done.

The scheduling and interaction of events is very important and needs to be extended. Events should be able to be scheduled relative to other events or internal clocks. "Object" should be able to accept messages to handle simple interactions. For example, one should be able to tell an "Object" to go forward until it runs into another "Object."

This view of programming as collections of actors, or a community of "little people," that send and receive messages from each other is very powerful.

Another improvement being considered is the ability to handle partial messages. For example, if an object receives a forward message without a number as the second word, then instead of the present response of printing out an error message, it should prompt the user with a question like, "How much should Fred go forward?"

Taking this idea one step further I plan to have a "help" button that can be pushed at any time. If one has typed only part of a message and then the help button, then the actor may be able to finish part or all of the message for the user. This feature will hopefully alleviate many of the problems of typing long names and messages that make the code more readable.

It should be clear by now that this system is far from complete and I welcome any suggestions or criticism.

I would suggest that anyone wishing more information look into *Director Guide,* M.I.T. AI Memo #482B, December 1979. □

Bibliography

1) Adler, M., "Understanding Peanuts Cartoons", *Progress in Perception,* Department of Artificial Intelligence Research Report No. 13, University of Edinburgh, December 1975.

2) Baecker, R., "Interactive Computer-Mediated Animation", MIT EE Ph.D. Thesis 1969 MAC-TR-61.

3) Goldberg, A., "Smalltalk and Kids — Commentaries", Learning Research Group, Xerox Palo Alto Research Center, 1974 Draft.

4) Goldstein, I.P., *Understanding Simple Picture Programs,* MIT AI Laboratory AI-TR-294, September 1974.

5) Goldstein, I., Leiberman, H., Bochner, H., Miller, M., "LLOGO: An Implementation of LOGO in LISP", MIT-AI Memo 307, March 4, 1975.

6) Goldstein, I., Papert, S., "AI, Language and the Study of Knowledge", MIT-AI Memo 337, 1976.

7) Hewitt, C., Smith, B., "Towards a Programming Apprentice", IEEE Transactions on Software Engineering SE-1, March 1975.

8) Kahn, K., "Three Interactions Between AI and Education", *Machine Representations Of Knowledge,* ed. Elcock, E., Michie, D., D. Reidel Publishing Company, Dordrecht, Holland, in press.

9) Minsky, M., "A Framework for Representing Knowledge", *The Psychology of Computer Vision,* ed. Winston, P., McGraw-Hill Book Company, New York, 1975.

10) Papert, S., "Teaching Children Thinking", MIT-AI Memo 247, October 1971.

11) Papert, S., "Teaching Children To Be Mathematicians vs. Teaching About Mathematics", MIT-AI Memo 249.

12) Parke, F.I., "Computer Generated Animation of Faces", Univ. of Utah Report CSc-72-120, June 1972.

13) Schank, R., *Conceptual Information Processing,* North Holland Publishing Company, 1975.

14) Schulz, C., "All This and Snoopy, Too", Fawcett Crest Publications, Inc., Greenwich, Conn. 1962.

15) Schulz, C., *Peanuts Jubilee — My Life and Art with Charlie Brown and Others,* Holt, Rinehart and Winston, New York, 1975.

16) Simmons, R., "The Clowns Microworld", *Theoretical Issues in Natural Language Processing,* ed. Schank, R. and Nash-Weber, B., June, 1975.

17) Smith, B. and Hewitt, C., "A Plasma Primer", MIT-AI Working Paper 92, October, 1975.

18) Sutherland, I., "Sketchpad: A Man-Machine Graphical Communication System", MIT Lincoln Tab TR-296, 1963.

Settle for More from Your TRS-80

BASIC Compiler. With TRS-80 BASIC Compiler, your Level II BASIC programs will run at record speeds! Compiled programs execute an average of 3-10 times faster than programs run under Level II. Make extensive use of integer operations, and get speeds 20-30 times faster than the interpreter.

Best of all, BASIC Compiler does it with BASIC, the language you already know. By compiling the same source code that your current BASIC interprets, BASIC Compiler adds speed with a minimum of effort.

And you get more BASIC features to program with, since features of Microsoft's Version 5.0 BASIC Interpreter are included in the package. Features like the WHILE . . .WEND statement, long variable names, variable length records, and the CALL statement make programming easier. An exclusive BASIC Compiler feature lets you call FORTRAN and machine language subroutines much more easily than in Level II.

Simply type in and debug your program as usual, using the BASIC interpreter. Then enter a command line telling the computer what to compile and what options to use.

Voila! Highly optimized, Z-80 machine code that your computer executes in a flash! Run it now or save it for later. Your compiled program can be saved on disk for direct execution every time.

Want to market your programs? Compiled versions are ideal for distribution.* You distribute only the object code, not the source, so your genius stays fully protected.

BASIC Compiler runs on your TRS-80 Model I with 48K and disk drive. The package includes BASIC Compiler, linking loader and BASIC library with complete documentation. $195.00.

*Microsoft royalty information for the sale of programs compiled with BASIC Compiler is available from Microsoft.

muMATH Symbolic Math System expands your TRS-80 beyond the limits of numerical evaluation to a much higher level of math sophistication.

Symbolic mathematics is muMATH's power. For the first time, algebra, trigonometry, calculus, integration, differentiation and more can be performed on a system smaller than an IBM 370. And in a fraction of the time you could do them manually.

Yet for all its power, muMATH is simple to use.

To perform a differentiation you could enter: ?DIF (A*X ↑ 3 + SIN(X ↑ 2),X);

In almost no time, the computer would reply with: @2*X*COS(X ↑ 2) + 3*A*X ↑ 2.

Or to add fractions: ?1/3 + 5/6 + 2/5 + 3/7;

The instantaneous answer: 419/210.

Or to perform a more difficult trigonometric expansion you enter: SIN(2*Y)*(4*COS(X)↑3—COS (3*X) + SIN (Y)*(COS(X+Y+#PI) — COS(X—Y));

Just a few seconds later, the computer replies: @4*SIN(Y)*COS(X)*COS(Y).

muMATH has virtually infinite precision with full accuracy up to 611 digits.

If you use math, you'll find countless ways to save time and effort with muMATH. It's a professional tool for engineers and scientists. A learning tool for students at any level from algebra to calculus.

And if you want to expand your capabilities even beyond the standard muMATH, the option is open. muSIMP, the programming language in which muMATH is written, is included in the muMATH package. A superset of the language LISP, muSIMP is designed especially for interactive symbolic mathematics and other artificial intelligence applications.

muMATH and muSIMP were written by The Soft Warehouse, Honolulu, Hawaii. Priced at $74.95, the package includes muMATH, muSIMP and a complete manual. It requires a Model I TRS-80 with 32K and single disk. muMATH for the Apple II Computer will be available later this year.

You can buy muMATH and BASIC Compiler at computer stores across the country that carry Microsoft products. If your local store doesn't have them, call us. 206-454-1315. Or write Microsoft Consumer Products, 400 108th Avenue N.E., Suite 200 Bellevue, WA 98004.

MICROSOFT CONSUMER PRODUCTS

CIRCLE 168 ON READER SERVICE CARD

3. Orderly Debugging (from La Vie des Animaux)

Same Screen, cont'd...

This has an important bearing on the problem of conceptual clarity — especially clarity in multi-process simulations. Multi-process systems are of increasing interest to computerists, for they are closer in spirit to the complex environments which we encounter in real life, and very different from the simple checkbook programs or Adventure games. So, in a Space War I have independent ships, missiles, heavenly bodies — each affects the others, all must be processed every time step. This is where Smalltalk (and other actor-based languages) make implementation so straightforward. The language itself contains constructs for multiple processes and, most importantly, communication structures between them. It is substantially easier in a language with multi-process programming facilities already built for it. Concepts like Class bootstrap the programmer into the bliss of specifying behaviors directly and without repetition. (In Basic, arrays of objects would be required, with explicit code to scan the arrays, pass parameters, test for conflict; all processing having to be under the programmer's elaborate and cumbersome control.) Actor languages also allow you to program faster and help you find those structures which preserve, and are sympathetic to, your own personal approach.

I came upon Smalltalk several years ago and was struck by the intelligence of its general philosophy — with one significant reservation. If one is interested in multi-object simulations for graphical display (a premise of Smalltalk's originators), then why am I forced to continue specifying my instructions in a linear, character-by-character, *typed* language?

Moreover, I want to make pictures with the computer, so why couldn't a system be made where the programs themselves were pictures (or at least two-dimensional things)? Might we not blur the distinction between programming and animating?

I was fortunate to be at a research laboratory at MIT which had both the facilities and the expertise to explore this idea. The result of this collaboration was a system called EOM, whose name has lost its historical meaning. *(Originally suggested by McCullough's "Embodiments of Mind"; see bibliography, — Ed.)* EOM was essentially an actor language, but one in which interconnection of actors by messages was visually configurable. Our implementation used a lightpen on a vector display with menus for programming and scripting animations.

The fundamental idea of EOM was that all programs are two-dimensional scripts, whose graphical nodes themselves stood for executable programs. Two aspects were both controlled from the screen: the data paths of the program and the graphics of the intended animation.

Our script convention of data flow is that lines drawn into the top of nodes are inputs, and lines from the bottom are outputs. Such links are the paths of data flow.

First let's look at a straight program example:

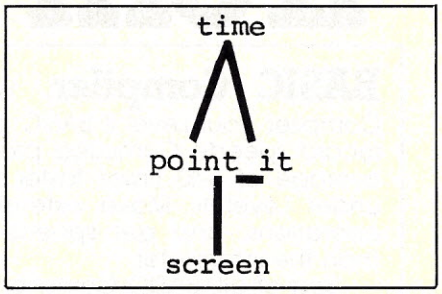


Figure 1.

The script in Figure 1 simply moves a point diagonally across the frame. The node *time* outputs a value — an ascending integer — as a function of absolute time in the sequence. (This value is essentially a frame number.) This value takes a split path to both the x and y input of *point it.* The object *point it* now sends two values (in this case, its inputs) to *screen,* an actor which displays it.

The real advantages for graphics are to come when we add in picture elements. Consider the following script.

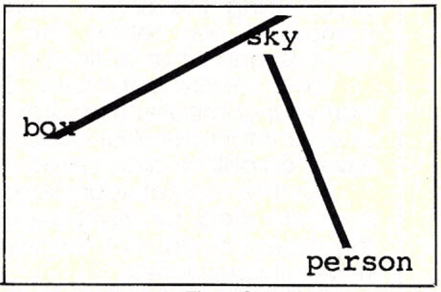


Figure 2.

This program will produce a box sailing across the frame. The outputs of *box* are essentially its shape and position, which change. These variables are repeatedly sent to *sky.* The position of the *word* "sky" indicates the highest point the box is to reach.

The box here is like a subroutine, which might be defined like this:

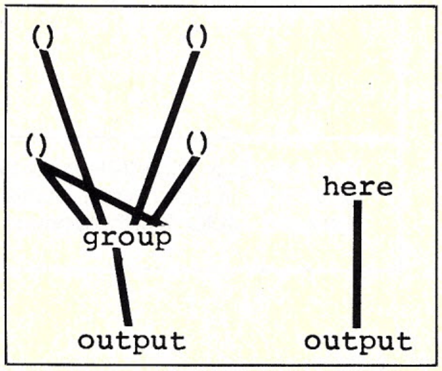
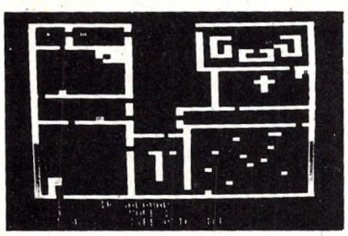


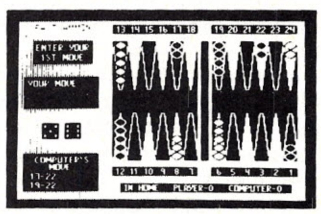
Figure 3.

Exciting, entertaining software for the Apple II and Apple II Plus*

If you liked "Invaders", you'll love ASTEROIDS IN SPACE by Bruce Wallace. Your space ship is traveling in the middle of a shower of asteroids. Blast the asteroids with lasers, but beware — big asteroids fragment into small asteroids! The Apple game paddles allow you to rotate your space ship, fire its laser gun, and give it thrust to propel it through endless space. From time to time you will encounter an alien space ship whose mission is to destroy you, so you'd better destroy it first! High resolution graphics and sound effects add to the arcade-like excitement that this program generates. Runs on any Apple II with at least 32K and one disk drive.
On diskette — **$19.95**

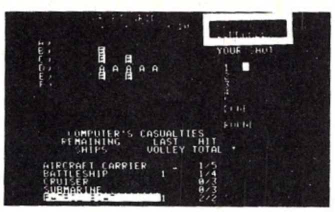


FRACAS™ by Stuart Smith. A fantastic adventure game like no other — up to eight players can participate in FRACAS at the same time. Journey in the land of FAROPH, searching for hidden treasure while warding off all sorts of unfriendly and dangerous creatures like the Ten Foot Spider and the Headless Horseman. You and your friends can compete with each other or you can join forces and gang up on the monsters. Your location is presented graphically and sound effects enliven the battles. Save your adventure on diskette or cassette and continue it at some other time. Requires at least 32K of RAM. Cassette: **$19.95** Diskette: **$24.95**



FASTGAMMON™ by Bob Christiansen. Sound, hi res color, and cartoons have helped maked this the most popular backgammon-playing game for the Apple II. But don't let these entertaining features fool you — FASTGAMMON plays serious backgammon. Requires at least 24K of RAM.
Cassette: **$19.95** Diskette: **$24.95**

BATTLESHIP COMMANDER™ by Erik Kilk and Matthew Jew. A game of strategy. You and the computer each start out by positioning five ships of different sizes on a ten by ten grid. Then the shooting starts. Place your volleys skillfully — a combination of logic and luck are required to beat the computer. Cartoons show the ships sinking and announce the winner. Sound effects and flashing lights also add to the enjoyment of the game. Requires at least 32K of RAM.
Cassette: **$14.95** Diskette: **$19.95**



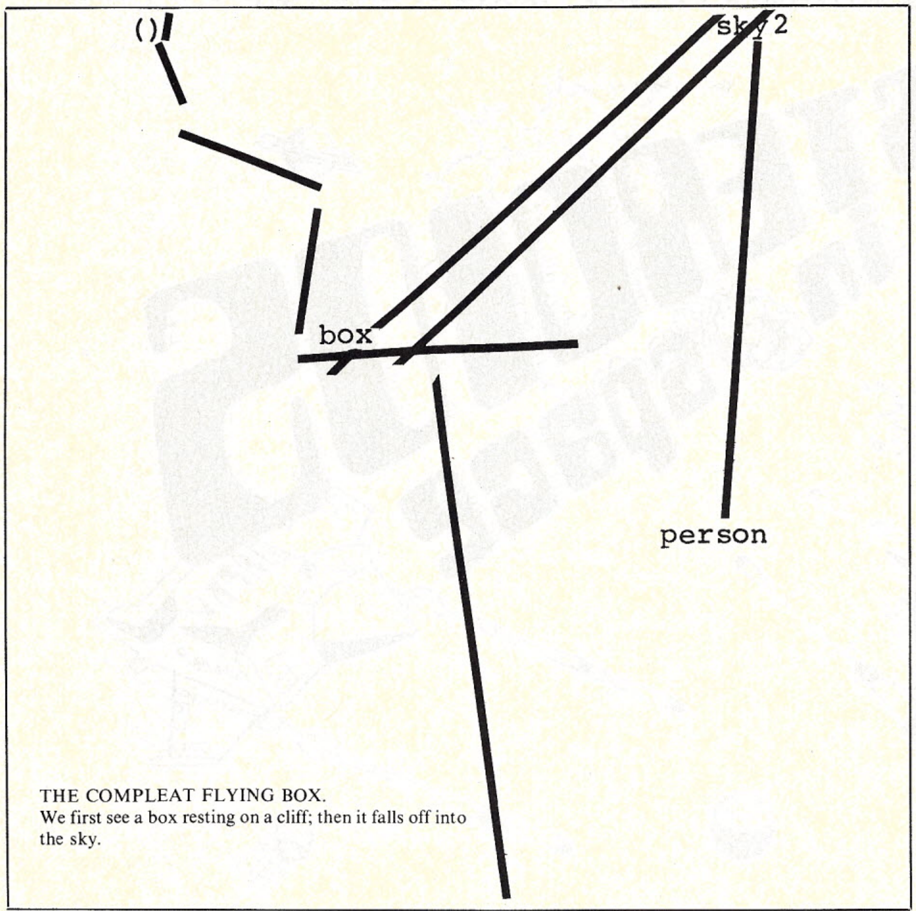
QS QUALITY SOFTWARE

6660 Reseda Blvd., Suite 105, Reseda, CA 91335

WHERE TO GET IT: Call us at (213) 344-6599 for the name of the Quality Software dealer nearest you. If necessary, you may order directly from us. MasterCharge and Visa cardholders may place orders by telephone and we will deduct $1 from orders over $19 to compensate for phone charges. Or mail your order to the address above. California residents add 6% sales tax. SHIPPING CHARGES: Within North America orders must include $1.50 for first class shipping and handling. Outside North America the charge for airmail shipping and handling is $5.00 — payable in U.S. currency.

*"Apple II" and "Apple II Plus" are trademarks of Apple Computer, Inc.

CIRCLE 231 ON READER SERVICE CARD



THE COMPLEAT FLYING BOX.
We first see a box resting on a cliff; then it falls off into the sky.

Figure 4.

Same Screen, cont'd...

The *( )* nodes indicate positions on the screen; their output is therefore their positions x,y. Thus to change the shape of the box, you may grab any of the *( )* nodes by lightpen and move where you like, defining the new shape.

The *group* node assembles the positions and outputs them from the subroutine. (The five connections to *group* complete the closed figure of four points and four lines.)

The *here* node happens to output a position also: in this case, the x,y position of the node *box* in Figure 2. This is passed as output out of the *box* subroutine, and linked to the input of *sky* (in Figure 3) which uses it to fix the initial position of the box which appears in the animated sequence.

Thus moving the position of the *box* node at the top level changes the initial position of the box in the animation — a simple change which does not require the usual calibrating and measuring: "The box is too far to the left by 10%, the x,y are 0 to 512 so we must add 50 units to the x position which is currently 120 so the new position is 170, type that in . . . " None of that nonsense — just grab the box and move it to its new position. I cannot imagine anything simpler.

The position of *sky* in the upper level is also significant, since it is programmed to determine the precise trajectory of the box. The *person* node is simply a metaphor, for it is defined thus:

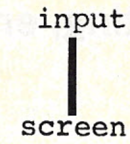


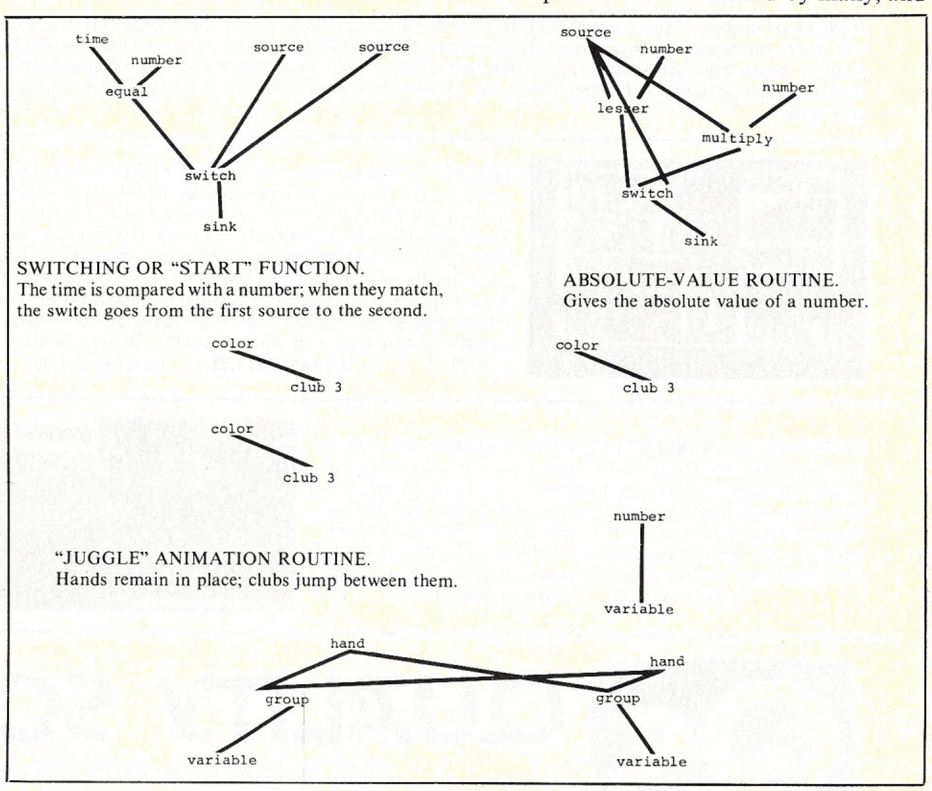
Figure 5.

This "person" takes all its inputs and displays them on the screen: I see this as answering the question of whether a tree falling in the forest makes a sound even if nobody is there to hear it.

The system was found by users to be most pleasing to interact with, and was extremely helpful in animation production. (We managed to film sequences for the science series NOVA, under the usual absurd production schedules, for which the system was superb.)

As an environment for education, EOM has the advantage that a student can perform many simulation experiments, knowing nothing about Cartesian coordinates or programming. Given a set of simulation models, all possible degrees of freedom can be expressed graphically as described above, making interaction simple. For the knowledgeable student, the models themselves can be manipulated, building on the uniform and extensible environment.

Alas, after many months of glory the system died when the hardware configuration was dismantled for newer research, and all that we learned must await another propitious time for further development. Its spirit is carried forward by many, and Henry Lieberman at the MIT Artificial Intelligence Laboratory has extended the concept substantially with his Tinker system. □



SWITCHING OR "START" FUNCTION.
The time is compared with a number; when they match, the switch goes from the first source to the second.

ABSOLUTE-VALUE ROUTINE.
Gives the absolute value of a number.

"JUGGLE" ANIMATION ROUTINE.
Hands remain in place; clubs jump between them.

NEW! TPM* for TRS-80 Model II
NEW! System/6 Package
Computer Design Labs

Z80* Disk Software

We have acquired the rights to all TDL software (& hardware). TDL software has long had the reputation of being the best in the industry. Computer Design Labs will continue to maintain, evolve and add to this superior line of quality software.
— Carl Galletti and Roger Amidon, owners.

Software with Manual/Manual Alone

All of the software below is available on any of the following media for operation with a Z80 CPU using the CP/M* or similar type disk operating system (such as our own TPM*).

for TRS-80* CP/M (Model I or II)
for 8" CP/M (soft sectored single density)
for 5¼" CP/M (soft sectored single density)
for 5¼" North Star CP/M (single density)
for 5¼" North Star CP/M (double density)

BASIC I

A powerful and fast Z80 Basic interpreter with EDIT, RENUMBER, TRACE, PRINT USING, assembly language subroutine CALL, LOADGO for "chaining", COPY to move text, EXCHANGE, KILL, LINE INPUT, error intercept, sequential file handling in both ASCII and binary formats, and much, much more. It runs in a little over 12 K. An excellent choice for games since the precision was limited to 7 digits in order to make it one of the fastest around. $49.95/$15.

BASIC II

Basic I but with 12 digit precision to make its power available to the business world with only a slight sacrifice in speed. Still runs faster than most other Basics (even those with much less precision). $99.95/$15.

BUSINESS BASIC

The most powerful Basic for business applications. It adds to Basic II with random or sequential disk files in either fixed or variable record lengths, simultaneous access to multiple disk files, PRIVACY command to prohibit user access to source code, global editing, added math functions, and disk file maintenance capability without leaving Basic (list, rename, or delete). $179.95/$25.

ZEDIT

A character oriented text editor with 26 commands and "macro" capability for stringing multiple commands together. Included are a complete array of character move, add, delete, and display function. $49.95./$15.

ZTEL

Z80 Text Editing Language - Not just a text editor. Actually a language which allows you to edit text and also write, save, and recall programs which manipulate text. Commands include conditional branching, subroutine calls, iteration, block move, expression evaluation, and much more. Contains 36 value registers and 10 text registers. Be creative! Manipulate text with commands you write using Ztel. $79.95/$25.

TOP

A Z80 Text Output Processor which will do text formatting for manuals, documents, and other word processing jobs. Works with any text editor. Does justification, page numbering and headings, spacing, centering, and much more! $79.95/$25.

MACRO I

A macro assembler which will generate relocateable or absolute code for the 8080 or Z80 using standard Intel mnemonics plus TDL/Z80 extensions. Functions include 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical operations, local and global symbols, chaining files, linking capability with optional linker, and recursive/reiterative macros. This assembler is so powerful you'll think it is doing all the work for you. It actually makes assembly language programming much less of an effort and more creative. $79.95/$20.

MACRO II

Expands upon Macro I's linking capability (which is useful but somewhat limited) thereby being able to take full advantage of the optional Linker. Also a time and date function has been added and the listing capability improved. $99.95/$25.

LINKER

How many times have you written the same subroutine in each new program? Top notch professional programmers compile a library of these subroutines and use a Linker to tie them together at assembly time. Development time is thus drastically reduced and becomes comparable to writing in a high level language but with all the speed of assembly language. So, get the new CDL Linker and start writing programs in a fraction of the time it took before. Linker is compatible with Macro I & II as well as TDL/Xitan assemblers version 2.0 or later. $79.95/$20.

DEBUG I

Many programmers give up on writing in assembly language even though they know their programs would be faster and more powerful. To them assembly language seems difficult to understand and follow, as well as being a nightmare to debug. Well, not with proper tools like Debug I. With Debug I you can easily follow the flow of any Z80 or 8080 program. Trace the program one step at a time or 10 steps or whatever you like. At each step you will be able to see the instruction executed and what it did. If desired, modifications can then be made before continuing. It's all under your control. You can even skip displaying a subroutine call and up to seven breakpoints can be set during execution. Use of Debug I can pay for itself many times over by saving you valuable debugging time. $79.95/$20.

DEBUG II

This is an expanded debugger which has all of the features of Debug I plus many more. You can "trap" (i.e. trace a program until a set of register, flag, and/or memory conditions occur). Also, instructions may be entered and executed immediately. This makes it easy to learn new instructions by examining registers/memory before and after. And a RADIX function allows changing between ASCII, binary, decimal, hex, octal, signed decimal, or split octal. All these features and more add up to give you a very powerful development tool. Both Debug I and II must run on a Z80 but will debug both Z80 and 8080 code. $99.95/$20.

ZAPPLE

A Z80 executive and debug monitor. Capable of search, ASCII put and display, read and write to I/O ports, hex math, breakpoint, execute, move, fill, display, read and write in Intel or binary format tape, and more! on disk

APPLE

8080 version of Zapple

NEW! TPM now available for TRS-80 Model II!

TPM*

A NEW Z80 disk operation system! This is not CP/M*. It's better! You can still run any program which runs with CP/M* but unlike CP/M* this operating system was written specifically for the Z80* and takes full advantage of its extra powerful instruction set. In other words its not warmed over 8080 code! Available for TRS-80* (Model I or II). Tarbell, Xitan DDDC, SD Sales "VERSA-FLOPPY", North Star (SD&DD), and Digital (Micro) Systems. $79.95/$25.

SYSTEM MONITOR BOARD (SMB II)

A complete I/O board for S-100 systems. 2 serial ports, 2 parallel ports, 1200/2400 baud cassette tape interface, sockets for 2K of RAM, 3-2708/2716 EPROM's or ROM, jump on reset circuitry. Bare board $49.95/$20.

ROM FOR SMB II

2KX8 masked ROM of Zapple monitor. Includes source listing $34.95/$15.

PAYROLL (source code only)

The Osborne package. Requires C Basic 2.
5" disks $124.95 (manual not included)
8" disks $ 99.95 (manual not included)
Manual $20.00

ACCOUNTS PAYABLE/RECEIVABLE (source code only)

By Osborne, Requires C Basic 2
5" disks $124.95 (manual not included)
8" $99.95 (manual not included)
Manual $20.00

GENERAL LEDGER (source code only)

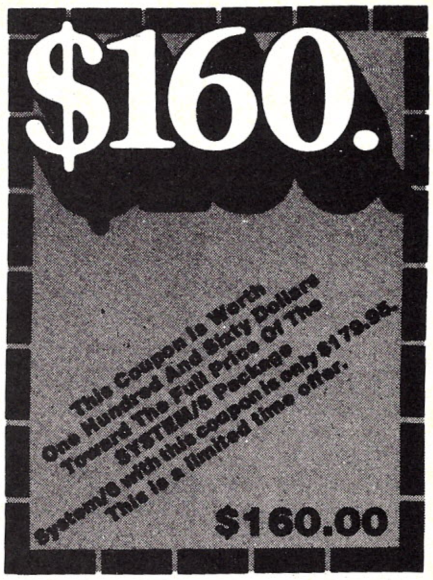
By Osborne. Requires C Basic 2
5" disks $99.95 (manual not included)
8" disks $99.95 (manual not included)
Manual $20.00

C BASIC 2

Required for Osborne software. $99.95/$20.

SYSTEM/6

TPM with utilities, Basic I interpreter, Basic E compiler, Macro I assembler, Debug I debugger, and ZEDIT text editor.
Above purchased separately costs $339.75
Special introductory offer. Only $179.75 with coupon!!



ORDERING INFORMATION

Visa, Master Charge and C.O.D. O.K. To order call or write with the following information.

1. Name of Product (e.g. Macro I)
2. Media (e.g. 8" CP/M)
3. Price and method of payment (e.g. C.O.D.) include credit card info. if applicable.
4. Name, Address and Phone number.
5. For TPM orders only: Indicate if for TRS 80, Tarbell, Xitan DDDC, SD Sales (5¼" or 8"), ICOM (5¼" or 8"), North Star (single or double density) or Digital (Micro) Systems.
6. N.J. residents add 5% sales tax.

Manual cost applicable against price of subsequent software purchase in any item except for the Osborne software.

For information and tech queries call
609-599-2146

For phone orders ONLY call toll free
1-800-327-9191
Ext. 676
(Except Florida)

OEMS

Many CDL products are available for licensing to OEMs. Write to Carl Galletti with your requirements.

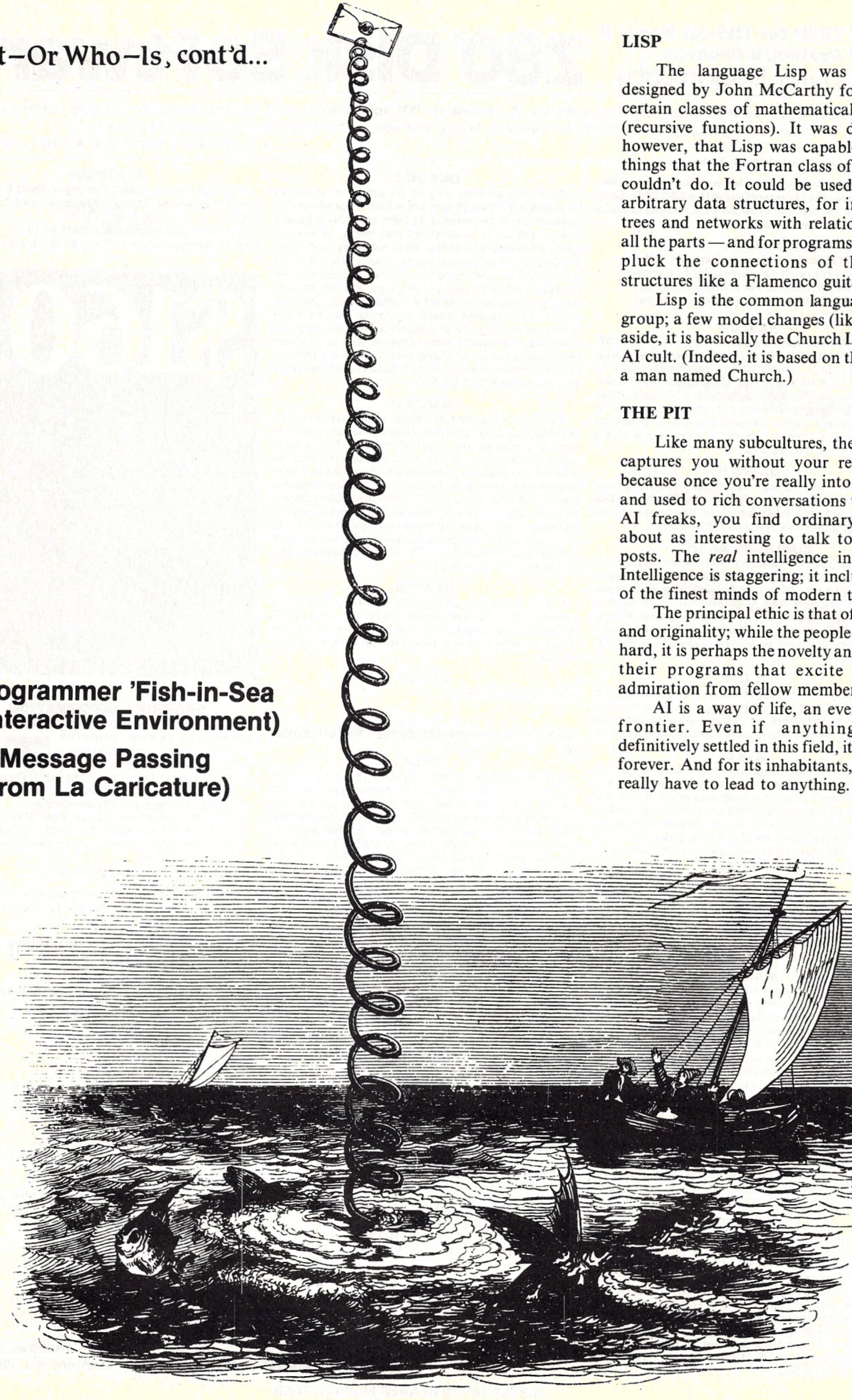
* Z80 is a trademark of Zilog
* TRS-80 is a trademark for Radio Shack
* TPM is a trademark of Computer Design Labs. It is not CP/M*
* CP/M is a trademark of Digital Research

Prices and specifications subject to change without notice.

DEALER INQUIRIES INVITED.

COMPUTER DESIGN LABS
342 Columbus Avenue
Trenton, N.J. 08629

CIRCLE 127 ON READER SERVICE CARD



4. Programmer 'Fish-in-Sea (In Interactive Environment)

Message Passing (from La Caricature)

LISP

The language Lisp was originally designed by John McCarthy for use with certain classes of mathematical problems (recursive functions). It was discovered, however, that Lisp was capable of doing things that the Fortran class of languages couldn't do. It could be used to create arbitrary data structures, for instance — trees and networks with relations among all the parts — and for programs that could pluck the connections of these data structures like a Flamenco guitarist.

Lisp is the common language of this group; a few model changes (like Scheme) aside, it is basically the Church Latin of the AI cult. (Indeed, it is based on the work of a man named Church.)

THE PIT

Like many subcultures, the AI world captures you without your realizing — because once you're really into the ideas, and used to rich conversations with other AI freaks, you find ordinary civilians about as interesting to talk to as fence-posts. The *real* intelligence in *Artificial* Intelligence is staggering; it includes some of the finest minds of modern times.

The principal ethic is that of brilliance and originality; while the people work very hard, it is perhaps the novelty and vision of their programs that excite the most admiration from fellow members.

AI is a way of life, an ever-receding frontier. Even if anything is ever definitively settled in this field, it will go on forever. And for its inhabitants, it doesn't really have to lead to anything. □

HAYDEN SOFTWARE...

New!

DATA MANAGER: A Data Base Management System and Mailing List

(Lutus) Do what the big machines do with your Apple II! This all-machine language program stores up to 96,000 alphanumeric characters on just one floppy disk. Powerful cursor-based editor facilitates easy information alteration in the data base. Program permits the user to sort on any key and subfiles on any search. Retrieve data in any combination of categories from up to 32,000 characters within one-half second. Choose between screen display or serial print-out (via the game paddle connector). Ideal for mailing lists, data banks, index files, or any other facts, files, or statistics that are begging to be organized. *#04909, Apple II Disk Version, $49.95.*

New!

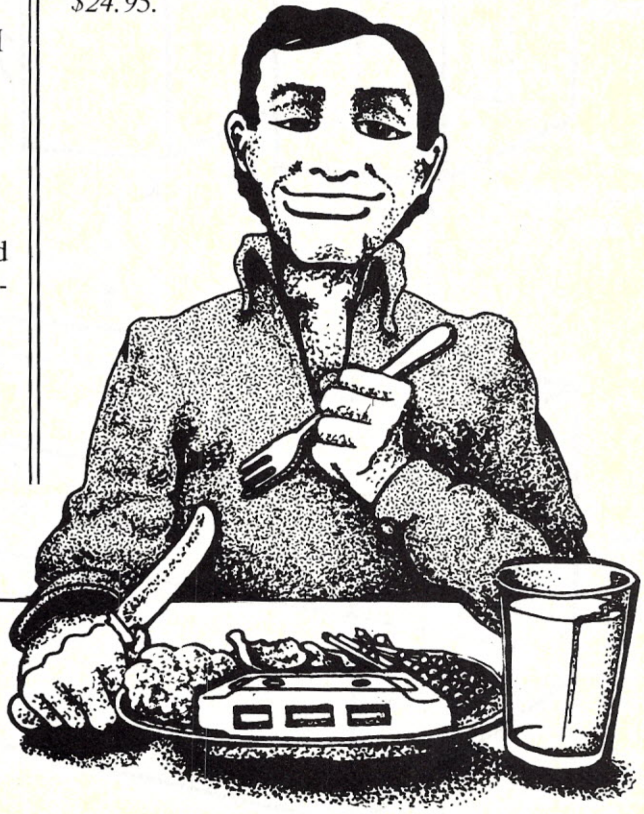
APPLE ™ ASSEMBLY LANGUAGE DEVELOPMENT SYSTEM

(Lutus) Enter the world of machine language programming with this brand new, winning, utility program. This very capable Assembler/ Editor/Formatter allows you to write and modify your language programs quickly and easily. Features a cursor-based editor, local and global labels, and disk-based macros which permit you to incorporate frequently used subroutines into any program. Formprint program lets you print a formatted listing of source and object files. *#04609, Apple II Disk Version, $39.95.*

New!

MICROCOMPUTER CIRCUIT ANALYSIS PROGRAM

(Savon) Minimize your time calculating the frequency response of a circuit with this new program. MCAP performs a linear voltage, impedence, or transfer impedence analysis of an electronic circuit. Enter the circuit description in a systematic nodal notation and the program then calculates, lists and plots the circuit's frequency response. MCAP readily analyzes circuits with up to 15 nodes — larger circuits can be divided into subsections for individual analysis. And, the circuits analyzed can contain any or all of the six types of components: resistors, capacitors, inductors, bipolar transistors, FETs, and Op-Amps. Educators in particular will find this a strong applications program for circuit analysis. *#04504, Apple II; #04501, PET; each Cassette $24.95.*



Available at your local computer store!

Hayden Book Company, Inc.

50 Essex Street, Rochelle Park, NJ 07662

or **Call Toll Free,** 24 hours a day,
(1-800-827-3777, ext. 302)*
TO CHARGE YOUR ORDER TO
Master Charge or Visa!
Minimum order is $10.00;
Customer pays postage and handling.
From Missouri, call
1-800-892-7655, ext. 302.

For those who prefer their ready-to-run programs

WELL DONE!

CIRCLE 199 ON READER SERVICE CARD

Bibliography, cont'd...

ment of Computer Science Technical Report 4-7, May 1968.

Carl Hewitt, "Viewing Control Structures as Patterns of Passing Messages." MIT AI Lab Memo 410, December 1976.

C. Hewitt, G. Attardi and H. Lieberman, "Security and Modularity in Message Passing." MIT AI Lab Working Paper 180, also in proceedings of First International Conference on Distributed Computing Systems, Huntsville, Alabama, October 1979.

Guy L. Steele and Gerald J. Sussman, "Constraints." MIT AI Lab Memo 502, November 1978.

Alan Borning, "Thinglab — A Constraint-Oriented Simulation Laboratory." Xerox Palo Alto Research Center; also Stanford Computer Science Dept. Report STAN-CS-79-746.

William A. Kornfeld, "A Synthesis of Language Ideas for AI Control Structures." MIT AI Working Paper #201, July 1980.

3. AI Actor Languages

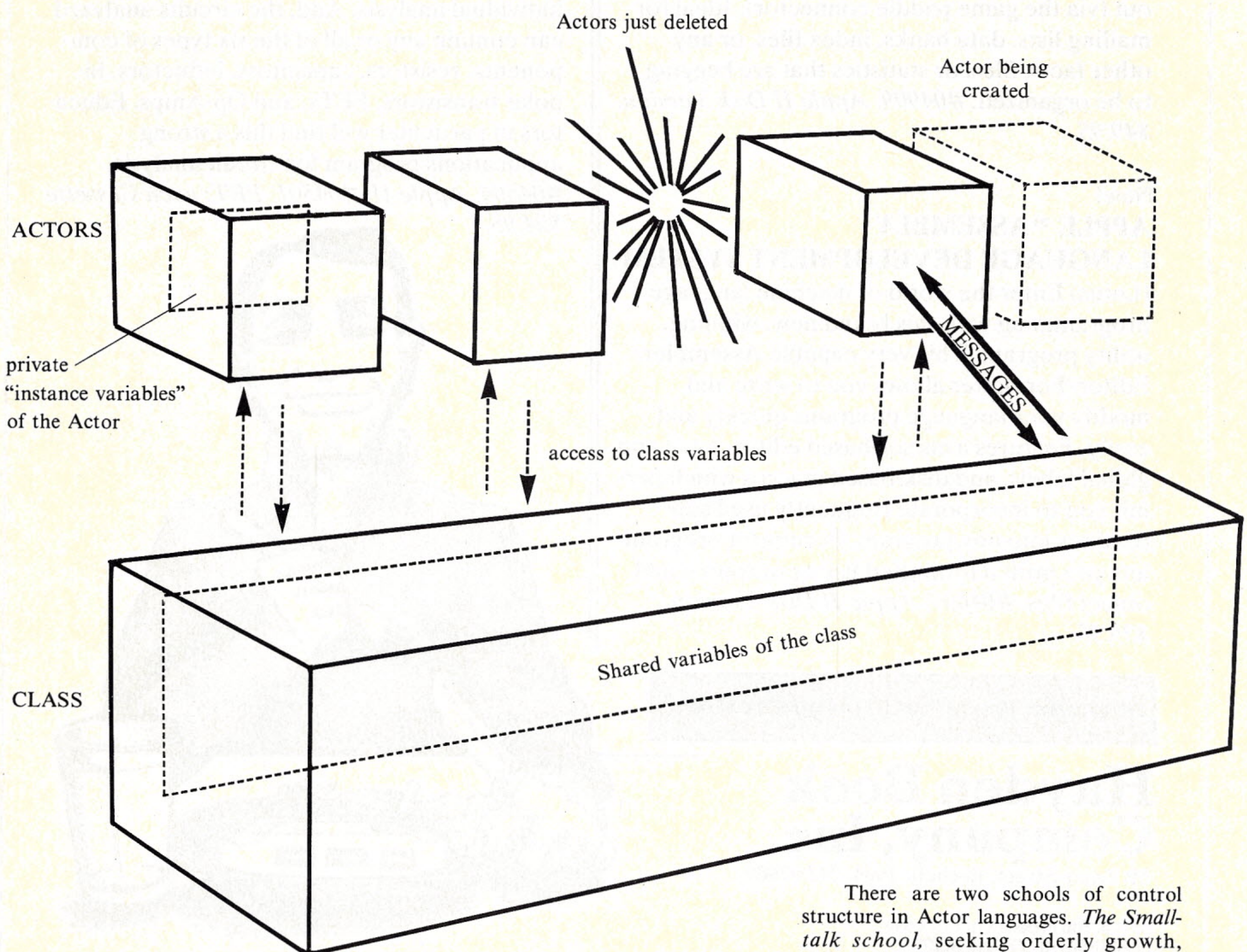
Carl Hewitt, "PLANNER — A Language for Manipulating Models and Proving Theorems in a Robot." Proceedings of the International Joint Conference on Artificial Intelligence, 1969.

Kenneth M. Kahn, "An Actor-Based Computer Animation Language." MIT AI Working Paper #120, February 1976. (Also complete in this issue of *Creative Computing.*)

D. C. Smith, "Pygmalion: A Creative Programming Environment." Stanford AI Memo #260, June 1975.

Paul Pangaro, Seth Steinberg, Jim Davis and Ben McCann, "EOM: A Graphically-Scripted, Simulation-Based Animation System." Architecture Machine Group, MIT, August 1977. Available for $4 from Paul Pangaro, 212 E 48th St., New York, NY 10017.

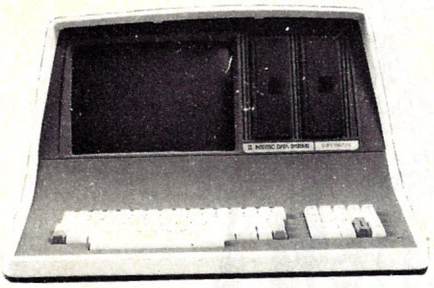
UNPLANNED NUMBERS OF ACTORS COME AND GO IN CORE MEMORY, WITHOUT CENTRAL CONTROL.



There are two schools of control structure in Actor languages. *The Smalltalk school,* seeking orderly growth, normal subroutining structure: essentially, each actor relinquishes control to the actor which originally summoned it. *The Hewitt school,* concerned with theoretical issues of artificial intelligence, allows "hairy" control structures of arbitrary topology — the ultimate rebuke to the structured programming ideology.

Try to beat our prices!

SUPERBRAIN® by Intertec



Self-contained computer with dual disks and two RS232C ports. Complete with CP/M® 2.2 and BASIC.

32K Double Density, List $2995 . **$2685**
64K Double Density, List $3345 .... $2883
64K MiniMicroMart-upgraded to Quad Density .... ***SPECIAL $3395***

VIDEO TERMINALS

NEW **EMULATOR (Intertec)**, List $895 $ 749
NEW INTERTUBE III List $895 ... **ONLY** $ 749
SOROC 120, List $995 ...... ***SPECIAL $ 729***
1Q140, List $1495 .... ***SPECIAL $1149***
PERKIN-ELMER 550, List $997 ...... $ 799
with anti-glare screen, $1027 .... $ 829
HAZELTINE 1410, List $900 .... $ 749
1420 .... $ 849
1500, List $1225 .... $ 879
1510, List $1395 .... $1089
1520, List $1650 .... $1389
ADDS R-20, List $995 .... $ 945
LEAR SIEGLER ADM3A, Assembled .. $ 849
TELEVIDEO 912C, List $950 .... $ 789
920C, List $1030 .... $ 849

PRINTERS

ANADEX DP-8000 .... $ 849
DP-9500, List $1650 .... $1399
DP-9501, List $1650 .... $1399
PAPER TIGER IDS-440, List $995 .... $ 895
w/graphics op., incl. buffer, $1195 ... $ 989
NEW IDS PAPERTIGER 460 . List 1295 **$1149**
NEW IDS PAPERTIGER 460G List $1394 **$1199**
NEC Spinwriters .... Call for Price
TELETYPE 43 KSR $1087
CENTRONICS
730-1 parallel interface .... ***NEW LOW $679***
737 parallel interface .. ***SUPER VALUE $849***
779 w/Tractor, List $1350 .... $1049
702 w/Tractor, VFU, List $2480 .... $1995
703 w/Tractor, VFU, List $2975 ...... $2395
704 w/Tractor, VFU, List $2350 ...... $1995
TI 810 Basic, List $1895 .... $1695
810/serial & Centronics-style parallel interface, List $1940 .... $1735
810 w/full ASCII (U/LC), Vertical Forms Control, Compressed Print .. $1895
TI 820 KSR, List $2165 .... $1895
TI 745 w/full ASCII, List $1695 .... $1399
COMPRINT 912 w/parallel interface .. $ 559
912 w/serial interface, List $699 ...... $ 589
AXIOM IMP I .... $ 699
MICROTEK, List $750 .... $ 675
OKIDATA Microline 80, List $949 649
Tractor Feed Option .... $ 99
RS232 Serial Interface .... $ 89

NORTH STAR HORIZON®

HORIZON 1 ASSEMBLED & TESTED
32K, Double Density, List $2695 .... $2279
32K, Quad Density, List $2995 .... $2539

HORIZON 2 ASSEMBLED & TESTED
32K, Double Density, List $3095 ...... $2619
32K, Quad Density, List $3595 .... $3049
48K, Double Density, List $3590 ...... $3039
48K, Quad Density, List $4090 .... $3469
64K, Double Density, List $3830 ...... $3239
64K, Quad Density, List $4330 .... $3669

LIMITED QUANTITY OF HORIZON 2 KITS AVAILABLE

FLOPPY DISK SYSTEMS

NORTH STAR MDS-A
Assembled, List $899 ...... ***SPECIAL $ 719***
Kit Version, List $799 .... $ 669
MORROW THINKER TOYS® **Discus 2D**, List $1199 .... **OUR PRICE** $1019*
Discus 2D, dual-drive, List $1994 ..... $1694*
Discus 2+2, A&T, List $1549 .... $1319*
Dual Discus 2+2, A&T, List $2748 ... $2335*
*Now includes CP/M® 2.2
MICROMATION Megabox, DD w/ 8" drives, 1-megabyte, List $2295 ..... $1949
2-megabyte, List $3095 .... $2629
MICROPOLIS 1041 MacroFloppy®
w/enclosure (no P.S.), List $695 ...... $ 625
1042 MacroFloppy w/case & AC P.S. . $ 709
1053 Dual MetaFloppy®, List $1895 ... $1695

VIDEO BOARDS

I/O Mapped
SD COMPUTER VDB-8024, kit, List $370 $319 †
Assembled, List $470 .... $ 399 †
XITEX SCT-100K, Kit .... **ONLY** $154.95
SCT-100A Assembled $174.95
SSM VB2 I/O, Kit, List $199 .... Call
Assembled & Tested, List $269 ..... Call
Memory Mapped
SSM VB1C, 16x64, Kit, List $179
Assembled & Tested, List $242
SSM VB3, 80-Char., 4MHz, Kit, List $485 Call
4 MHz, A&T, List $565 .... Call
INTERSYSTEMS, 16x64, A&T, List $165 $149

ESCON CONVERSION FOR IBM SELECTRIC

Complete w/microprocessor controller and power supply. Factory built. User installs solenoid assembly or it can be done at Escon factory at nominal cost.
Parallel (TRS-80, Sorcerer, etc.), $575 $514
RS232 Standard Serial, List $599 534
IEEE-488 (for PET), List $660 584
TRS-80 Cable 25

CALIFORNIA COMPUTER SYSTEMS

Z80 CPU BOARDS List $299 .... $269
DISK CONTROLLER 2422 List $399 .... $359
32 CASE STATIC List $710 .... $599
64K DYNAMIC BOARD List $699 .... $589

CPU BOARDS

(assembled unless noted)

NORTH STAR Z80A (ZPB-A/A), $299 $254
CROMEMCO 4 MHz (ZPU-W), List $395 $335
4 MHz (SCC-W), List $450 .... $382
INTERSYSTEMS (formerly Ithaca Audio)
new Series II Z-80, 4 MHz, List $395 ... $349
SSM CB1 8080 A&T, List $252 ....
CB1A Kit, List $183 ....
CB2 Z-80, A&T, List $344 ....
CB2 Kit, List $260 ....
DELTA Z-80, with I/O $289
SD SBC-100, List $350 .... $298
SBC-100 Kit, List $295 .... $250
SBC-200, List $400 .... $332
SBC-200 Kit, List $320 .... $272

MEMORY BOARDS

32K SD ExpandoRAM Kit
ONLY $249†
ONLY $159 without RAM chips

NORTH STAR 16K Dynamic RAM Board, A&T (RAM-16-A/A), List $499 .... $420
16K Kit Version, List $449 ... **SPECIAL** $299
32K A&T (RAM-32/A), List $739 .... $620
32K Kit, List $669 .... **SPECIAL** $499
CROMEMCO 16KZ-W, List $495 ...... $419
64KZ-W, List $1795 1485
MEASUREMENT SYSTEMS & CONTROLS
(Guaranteed performance, incl. labor/parts 1 yr)
DM6400 64K Board w/all 64K, $795 ... $659
DM4800 with 48K, List $695 .... $589
DM3200 with 32K, List $595 .... $509
DMB6400 64K Board w/all 64K $859
DMB4800 with 48K $789
MORROW SuperRAM — all static, all A&T
16K, 4 MHz or 2 MHz, List $349 .... $299
32K, 4 MHz, List $699 .... $629
16K Memory Master, List $399 .... $339
24K Memory Master, List $549 .... $465
INTERSYSTEMS (formerly Ithaca Audio)
8K Static 2 MHz, A&T, List $165 ...... $149
8K Static 4 MHz, A&T, List $195 ...... $176
16K Static 2 MHz, A&T, List $475 ..... $427
16K Static 4 MHz, A&T, List $495 ..... $445
64K Dynamic, List $995 .... $895
CALIFORNIA COMPUTER
16K Static, A&T, List $349.95 .... $259

FLOPPY DISK CONTROLLER BOARDS

NORTH STAR, DD, Assembled, List $499 .... $399
MORROW Disk Jockey 1, A&T ($213) . $189
Disk Jockey 2D, A&T, List $479 .... $429
SD Versafloppy 1, Kit, List $250 .... $212 †
Versafloppy II, DD Kit, List $350 ...... $297 †
Versafloppy II, DD, A&T, List $430 .... $365 †
DELTA double density A&T ($385) $345
CONDUCTOR, double density A&T ... $269
INTERSYSTEMS FDC-2, A&T, $495 ... $439
MICROMATION Doubler, DD, A&T ... $399
TARBELL Floppy Disk Interface Kit $199
double density, A&T, List $495 .... $444

NEW CROMEMCO DOUBLE DENSITY DISK CONTROLLER

List $595 ...... **OUR PRICE $505**

SHIPPING AND INSURANCE: Add $2.50 for boards, $6 for Selectric Converter or Floppy Disk Drives, $7.50 for Floppy Disk Systems, $15 for Horizon. SHIPPED FREIGHT COLLECT: SuperBrain, Centronics and T.I. printers. Contact us for shipping information on other terminals and printers.
Above prices reflect a 2% cash discount (order prepaid prior to shipment). Add 2% to prices for credit card orders, C.O.D.'s, etc. Prices are subject to change and offers subject to withdrawal without notice.
— WRITE FOR FREE CATALOG —

MiniMicroMart, Inc.

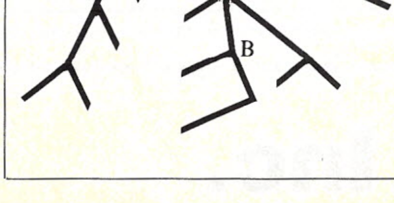
1618 James Street, Syracuse NY 13203 (315) 422-4467 TWX 710-541-0431

CIRCLE 180 ON READER SERVICE CARD

Bibliography, cont'd...

Some Actor languages (including Smalltalk 76) have a feature called "inheritance" whereby Actors may be defined as special cases of one another.

This involves a tree of classes. If class A is an ancestor of class B, then class A is a superclass of class B, and B is a subclass of A. A class inherits all of the characteristics of its superclasses. This turns out to be an excellent organizational mechanism for putting different types of behavior at different levels of abstraction. You can define all kinds of Bs that have all the properties of A — and then special ones of their own.



Henry Lieberman and Carl Hewitt, "A Session with Tinker." MIT AI Lab Memo #577, September 1980.

4. Smalltalk

Alan Kay, "A Personal Computer for Children of All Ages," Proceedings of the ACM National Conference, August 1972.

Alan Kay, "Microelectronics and the Personal Computer," *Scientific American,* September 1977, 230-244.

Alan Kay and Adele Goldberg, "Personal Dynamic Media," *IEEE Computer* March 1977, 31-41.

Scott Warren and Dennis Abbe, "Introducing Rosetta Smalltalk." *Datamation,* May 1980, 144-158.

Adele Goldberg and Alan Kay (eds.), *Smalltalk-72 Instruction Manual.* Xerox Palo Alto Research Center, 1976.

Rosetta Smalltalk Prototype Language Manual. Rosetta, Inc., 5925 Kirby Suite 215, Houston, TX 77005.

Daniel H. H. Ingals, "The Smalltalk-76 Programming System: Design and Implementation." *Conference Record of the Fifth Annual Symposium on Principles of Programming Languages,* Tucson, Arizona, January 1978, 9-16.

5. New, Obscure and Important Topics in Computer Languages

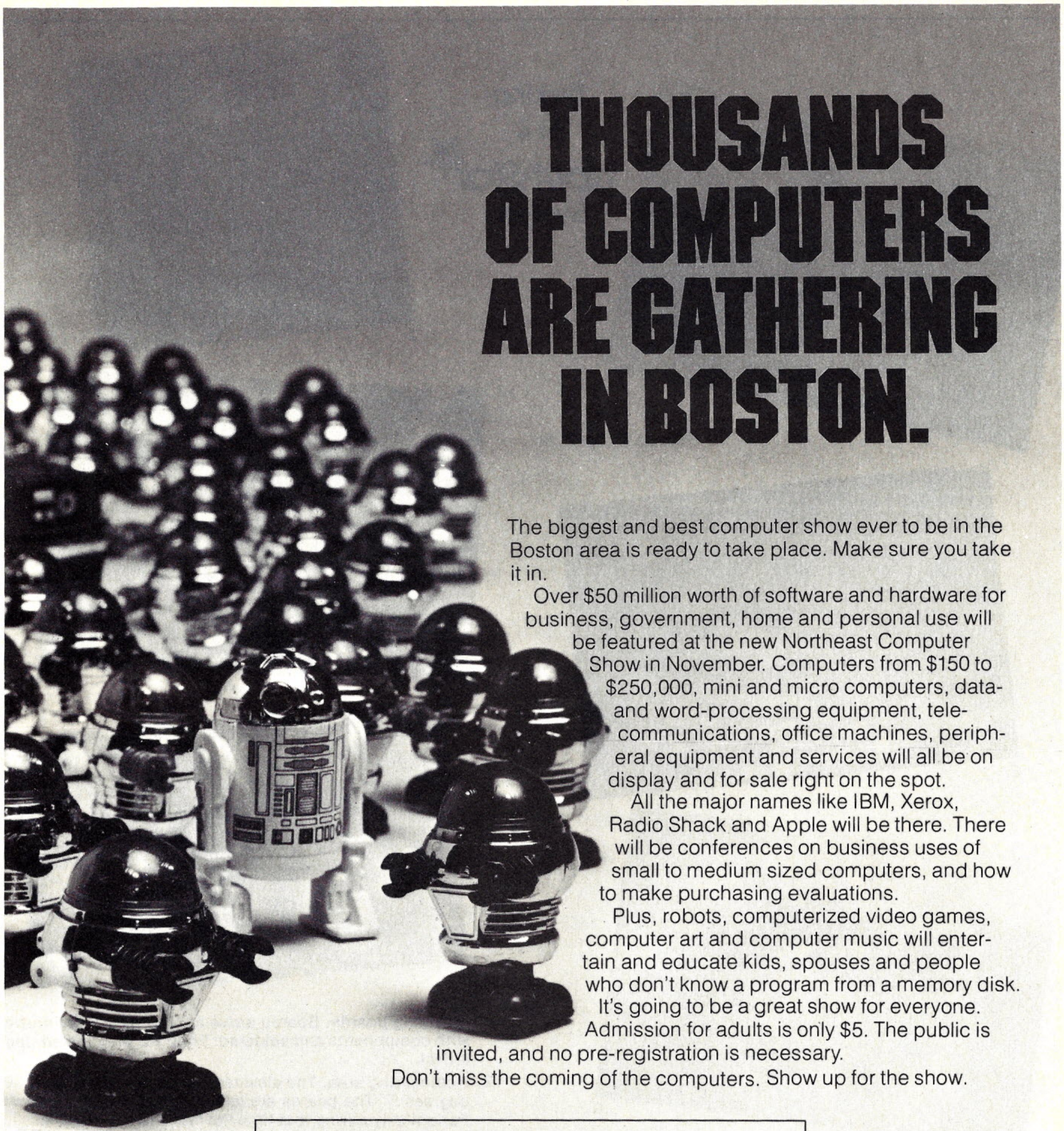
John Backus, "Can Programming Be Liberated from the von Neumann Style? A functional style and its algebra of programs." *Communications of the ACM,* August 1978, 613-641.

Kenneth E. Iverson, "Operators." *ACM Transactions on Programming Languages and Systems,* October 1979, 161-176.

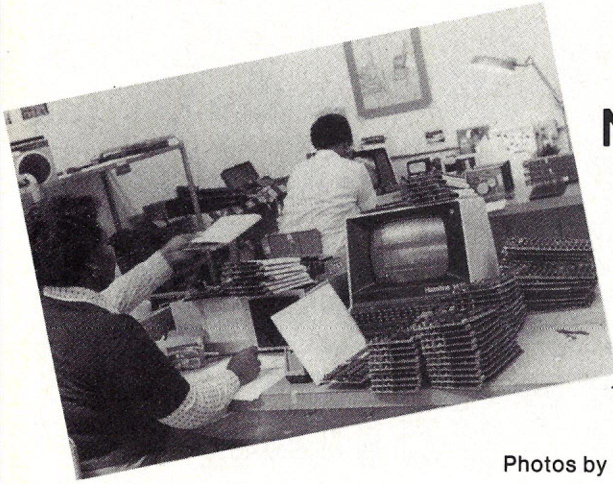
John R. Levine, "Why A Lisp-Based Command Language." *ACM SIGPLAN Notices,* May 1980, 49-53.

Marc A. Kaplan and Jeffrey D. Ullman, "A Scheme for the Automatic Inference of Variable Types." *Journal of the ACM,* January 1980, 128-145.

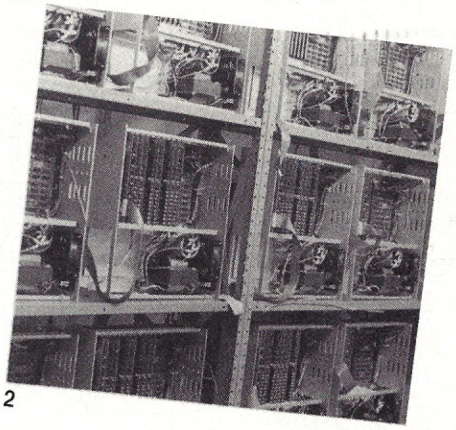
D. P. Friedman and D. S. Wise, "Cons Should Not Evaluate Its Arguments," in S. Michaelson and R. Milner (eds.), *Automata of Languages and Programming.* Edinburgh University Press, 1976, 257-284. □



Making of a NorthStar System

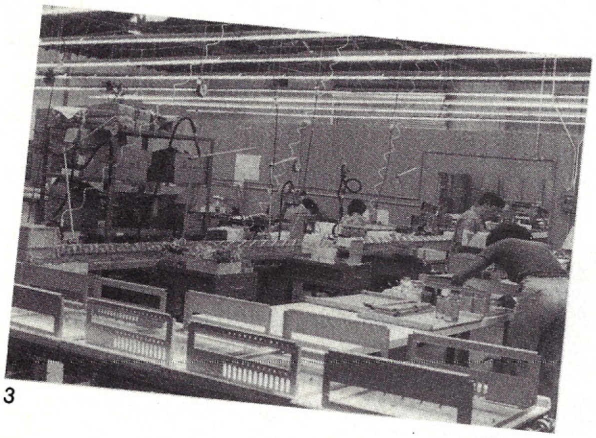


1



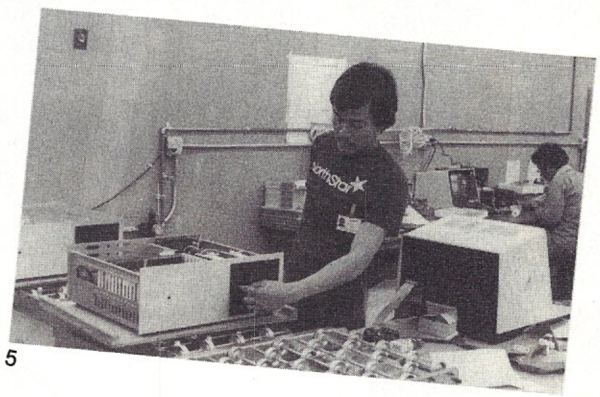
2

Photos by David Ahl.

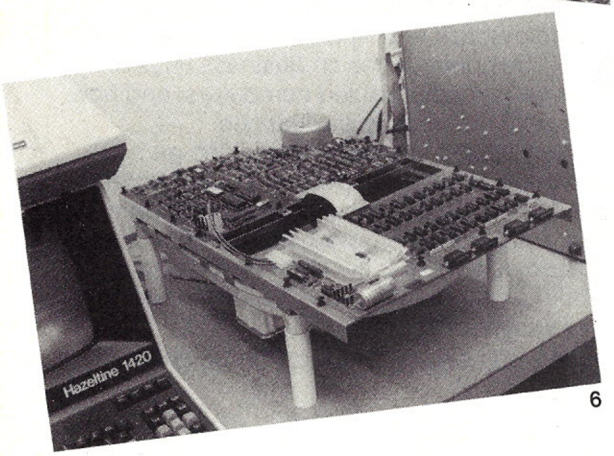


3

4



5



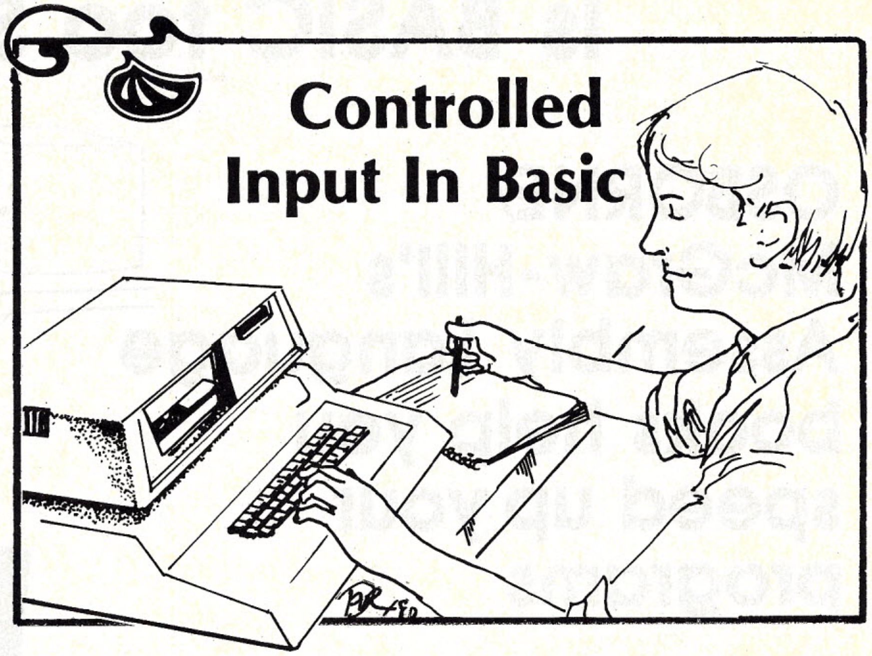
6

7

1. Incoming boards. Boards arrive at NorthStar fully stuffed with components and soldered. They are then tested upon arrival.
2. Heat testing area. The elevated temperature test is at 106 degrees F. The boards are left in this room for 48 hours. Periodically during that time the machines are tested.
3. Panoramic view of the Assemble line. (Just after lunch hour, the line was not fully operational.) Testing is first done on individual subassemblies rather than by complete systems, therefore when a system is being assembled the probability of failure is much lower. The final system itself is also tested.
4. Across the assembly line. The hanging plants and those being prepared to the left are not a NorthStar product.
5. Systems test. When a completed system comes out of production, it is configured for an individual customer and tested completely as a system.
6. Hard disk prior to being mounted in the cabinet. Hard disks are guaranteed for an error rate of no greater than 1 bit in 10^{10} bits. In terms of operation, it must run 36 hours absolutely error free.
7. Final test. Final system configuration is tested as it will be delivered to the customer.

Robert Montgomery

Controlled Input In Basic



Input of data from the keyboard is easy in Basic, isn't it? You just write "INPUT N", the computer responds with a question mark, and the operator provides an answer. Let's look at some of the things which are wrong with this method and how to fix them.

Let's assume that you're writing a program in Basic* to find out where your money goes. The way you plan it, you'll enter each check you write, and then be able to find out how much you paid to whom and for what. You set up a coding system, so that each household account has a number, as well as each of the persons you sent money to. Since you plan to record the entries on tape (or print a list as you go along), you'll also want the check number and the date, as well as the amount and the payee and account numbers.

You envision a screen display of all of the data for each transaction, and you want to be able to check the entries and correct errors before the data are stored or processed. Figure 1 shows how this screen might look just after you answer "N" to "CORRECTIONS (Y OR N)?"

A professional job of programming should stand up to the test of an unskilled operator.

```
          TRANSACTIONS FOR THE MONTH OF AUGUST 1980

DATE? 23

TRANSACTION NUMBER? 1232

AMOUNT? 56.24

PAYEE NUMBER? 71

ACCOUNT NUMBER? 32

CORRECTIONS (Y OR N)? N
```

Figure 1 '*TRS-80 Level II Basic is the dialect used in the illustrations.

As shown in Figure 2, the program segment which produced this result uses "INPUT" to obtain the required data, all of which are entered as simple variables.

```
100 CLS
110 PRINT TAB(10) "TRANSACTIONS FOR THE MONTH OF ";M$;Y$
120 PRINT
150 INPUT "DATE"; D
160 PRINT
200 INPUT "TRANSACTION NUMBER"; T
210 PRINT
300 INPUT "AMOUNT"; M
310 PRINT
400 INPUT "PAYEE NUMBER"; P
410 PRINT
500 INPUT "ACCOUNT NUMBER"; A
510 PRINT
600 INPUT "CORRECTIONS (Y OR N)"; A$
610 IF A$="Y" THEN 100
620 IF A$ <> "N" THEN GOSUB 1000 : GOTO 600
630 '''' PROCESSING STEPS FOR EACH TRANSACTION
640 GOTO 100 : '''' ENTER NEXT TRANSACTION
```

Figure 2

The display doesn't look too bad. But let's see what happens when the operator makes mistakes. Look at Figure 3.

```
          TRANSACTIONS FOR THE MONTH OF AUGUST 1980

DATE? 32

TRANSACTION NUMBER? 1232.15

AMOUNT? 56.24

PAYEE NUMBER? JONES PHARMACY
?REDO
PAYEE NUMBER? 71

ACCOUNT NUMBER?

CORRECTIONS (Y OR N)? N
```

Figure 3

There are now several problems, not all of which are visible.

1. Obviously, the date can't be "32". Equally obviously, you can write an error-trapping routine which will go tilt if D is less than 1 or more than 31. If you want to be extra sure, you can even write a "30 days hath September . . . " routine. Other data are also subject to logical limits.

2. The transaction number can't be fractional (not in my checkbook, at least). And such entries are easy to make if the operator thinks he's entering the amount. You should add a routine which calls an error message if the transaction number, the date, the account number or the payee number is not an integer.

3. The operator made another easy error in entering the payee name rather

Robert Montgomery, 67 Turtle Back Rd. West, New Canaan, CT 06840.

Input, cont'd. . .

than the payee number. The system error message "REDO" is already available to trap this one, for names won't be accepted when the computer expects a number. The problem here is that "REDO" printed on the next line and messed up your elegant display. The answer to this one is to enter all data as string variables, and then convert them to numbers as required. That way, you can control where and when error messages will print.

4. There's another argument in favor of using strings. Sometimes, you may want an alphabetic response in answer to "DATE?" For example, you need an exit routine, in order to go on to other things when all transactions have been entered. Answering "E" (for exit) in response to "DATE?" might be the best way to do this. You may also want the convenience of answering "N" (for no change) for any datum, if the answer is the same as the last transaction, or if you are making corrections. And you might want to permit an answer like "+" when the date or the transaction number is to be incremented by one from the last entry.

5. Because your error messages mess up the screen, just like "REDO", you want to control exactly where each question prints. And, if an error message is displayed, you want the screen restored to its previous condition.

Because your error messages can otherwise mess up the screen you want to control exactly where each question prints. And, if an error message is displayed, you want the screen restored to its previous condition.

6. When entering "ACCOUNT NUMBER?", the operator hit the "ENTER" key without entering a number (or, possibly, got a keybounce when entering "PAYEE NUMBER?"). Either way, the account number is whatever it was in the last transaction — and that number doesn't show on the screen. This condition should also generate an error message.

Figure 4 shows a modified program segment designed to meet these requirements.

We've added a lot of code to a simple program, but let's see what has been accomplished.

1. Each datum is now entered as a string — A$ is used repetitively for this purpose. It is nulled (A$ = "") before each use, so that its previous meaning is erased. If A$ = "N", the previous value

```
100 CLS
110 PRINT TAB(10) "TRANSACTIONS FOR THE MONTH OF ";M$;Y$
120 PRINT
150 A$ = "" : P = 128
160 PRINT @ P, "DATE"; : INPUT A$
170 IF A$ = "E" THEN 2000 ELSE IF A$ = "N" THEN 200 ELSE IF A$ = "+"
    THEN D = D + 1 : GOTO 180 ELSE D = VAL(A$)
180 IF D < 1 OR D >31 OR D <> INT(D) THEN GOSUB 1000 : GOTO 150
200 A$ = "" : P = 256
210 PRINT @ P, "TRANSACTION NUMBER"; : INPUT A$
220 IF A$ = "N" THEN 300 ELSE IF A$ = "+" THEN T = T + 1 : GOTO 230
    ELSE T = VAL (A$)
230 IF T <1 OR T >9999 OR T <> INT(T) THEN GOSUB 1000 : GOTO 200
300 A$ = "" : P = 384
310 PRINT @ P, "AMOUNT"; : INPUT A$
320 IF A$ = "N" THEN 400 ELSE M = VAL(A$)
330 IF M >= 1E6 OR (M = 0 AND A$ <> "0") THEN GOSUB 1000 : GOTO 300
400 A$ = "" : P = 512
410 PRINT @ P, "PAYEE NUMBER"; : INPUT A$
420 IF A$ = "N" THEN 500 ELSE N = VAL(A$)
430 IF N < 1 OR N > 99 OR N <> INT(N) THEN GOSUB 1000 : GOTO 400 ELSE
    IF P$(N) = "" THEN GOSUB 1000 : GOTO 400
500 A$ = "" : P = 640
510 PRINT @ P, "ACCOUNT NUMBER"; : INPUT A$
520 IF A$ = "N" THEN 600 ELSE A = VAL(A$)
530 IF A <1 OR A > 99 OR A <> INT(A) THEN GOSUB 1000 : GOTO 500 ELSE
    IF A$(A) = "" THEN GOSUB 1000 : GOTO 500
590 PRINT
600 INPUT "CORRECTIONS (Y OR N)"; A$
610 IF A$="Y" THEN 100
620 IF A$ <> "N" THEN GOSUB 1000 : GOTO 600
630 '''' PROCESSING STEPS FOR EACH TRANSACTION
640 GOTO 100 : '''' ENTER NEXT TRANSACTION

1000 PRINT @ P, CHR$(30); "ERROR -- PLEASE REENTER";
1010 FOR I = 1 TO 500
1020 NEXT
1030 PRINT @ P, CHR$(30)
1100 RETURN
```

Figure 4

of the numerical datum is unaffected, because an "N" results in bypassing the statement in which the numerical datum is set equal to VAL(A$). A$ = "E", in answer to "DATE?", results in exit from the transaction input routine, and A$ = "+", in answer to "TRANSACTION NUMBER?" or "DATE?" results in incrementing the previous datum by one, but still subject to the error check.

2. An error message subroutine begins at Line 1000. Note that it prints at position "P", where the erroneously-answered question was displayed, and that it first erases the line on which the message will print, using CHR$(30) for this purpose. It is displayed while a variable "K" loops uselessly from 1 to 500 and then erases itself.

3. Errors are recognized in the following cases:

a. *DATE* D less than 1, or D more than 31, or D not an integer. Since A$ was nulled, "ENTER" without an entry results in VAL(A$) = 0, which is less than 1, and an additional error check for this condition is unnecessary.

b. *TRANSACTION NUMBER* T less than 1, or more than 9999, or T not an integer. You'll want to change this if you must provide for transactions numbered past 9999.

c. *AMOUNT* M greater than or equal to $1,000,000 (1E6). Note also, however, that 0 could be a valid answer. Accordingly, an error message is also triggered if M = 0 *and* A$ is not equal to "0". As A$ would not be equal to "0" unless it were still a null, or a response like "XXX" were entered.

d. *PAYEE NUMBER* The illustration assumed that payee names are recorded in a string array called P$, which can contain 99 names but may contain less. Thus, an error occurs, first, if N is less than 1 or more than 99, or is not an integer. If the entry passes this test, an error message still occurs if P$(N) = "", which is to say that this particular name is not being used. Note that these tests must be separated and conducted in this order. Otherwise, N = 123, say, could result in a "BS ERROR" and a program crash when the computer finds that P$(123) is not an allowable subscript.

e. *ACCOUNT NUMBER* An assumption is again made that there are 99 accounts, not all of which are named. Thus, you have an error if A is less than 1, more than 99, or not an integer, and also if A$(A) = ""

How does the screen look now? Exactly the same, if there are no input errors. If mistakes were made on entry, the display is not disturbed. However, there are still some problems. The screen is not as easy to read as it might be when the time comes to check it over before answering "CORRECTIONS (Y OR N)?" Printing the data justified to the right margin will help, and it would be even better if Account Number 32 were printed as

```
          TRANSACTIONS FOR THE MONTH OF AUGUST 1980

DATE                                            23 AUGUST 1980

TRANSACTION NUMBER                                        1232

AMOUNT                                          $        56.24

PAYEE NAME                                      JONES PHARMACY

ACCOUNT NAME                              DRUGS AND MEDICINES

CORRECTIONS (Y OR N)? N
```

Figure 5

"DRUGS AND MEDICINES", rather than repeating what might have been an erroneous entry. The same thing applies to the Payee; "JONES PHARMACY" would be a lot more useful in checking for errors than repeating Payee Number 71.

Figure 5 shows the screen as it should be for easier checking. Note that it will not be changed by the use of "N" or "+".

Note that "ACCOUNT NUMBER?" and "PAYEE NUMBER?" change to "ACCOUNT NAME" AND "PAYEE NAME", through the use of carefully placed "PRINT CHR$(30)" statements in Lines 190, 240, 340, 440 and 540; that "N" answers result in branching to these lines, and that the TAB placement is computed where the length of the string is variable, as in Lines 440 and 540.

Figure 6 shows the program segment in its final form.

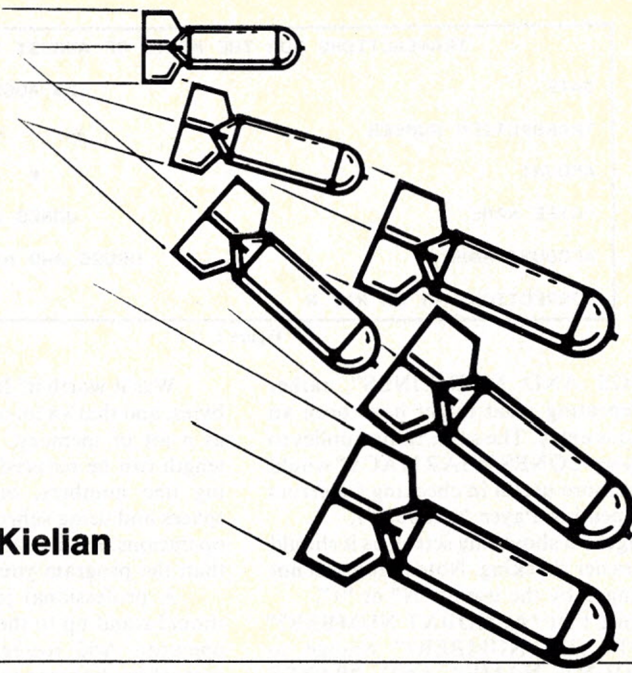
Was it worth it? It runs to almost 1100 bytes, and that's a lot of key strokes, as well as a lot of memory. First, of course, its length can be reduced by half by combining line numbers, omitting unnecessary spaces and using subroutines for repetitive operations. But it's still a lot more complex than the program you saw in Figure 2.

A professional job of programming should stand up to the test of an unskilled operator. And *you're* an unskilled operator when it comes to input, proficient as you may be as a programmer. Professional displays are not just a matter of pride; they increase operator confidence and pride in input accuracy. And almost any error trap you can devise is a sound investment; errors caught at the input stage are easy to fix. They're another matter when they survive to invalidate your final analytical reports. □

Figure 6

```
100 CLS
110 PRINT TAB(10) "TRANSACTIONS FOR THE MONTH OF ";M$;Y$
120 PRINT
150 A$ = "" : P = 128
160 PRINT @ P, "DATE"; : INPUT A$
170 IF A$ = "E" THEN 2000 ELSE IF A$ = "N" THEN 190 ELSE IF A$ = "+"
    THEN D = D  + 1 : GOTO 180 ELSE D = VAL(A$)
180 IF D < 1 OR D >31 OR D <> INT(D) THEN GOSUB 1000 : GOTO 150
190 PRINT @ P + 4, CHR$(30); TAB(47 - LEN(M$)) USING "##"; D; :
    PRINT " "; M$; Y $
200 A$ = "" : P = 256
210 PRINT @ P, "TRANSACTION NUMBER"; : INPUT A$
220 IF A$ = "N" THEN 240 ELSE IF A$ = "+" THEN T = T + 1 : GOTO 230
    ELSE T = VAL  (A$)
230 IF T <1 OR T >9999 OR T <> INT(T) THEN GOSUB 1000 : GOTO 200
240 PRINT @ P + 18, CHR$(30); TAB(51) USING "####"; T
300 A$ = "" : P = 384
310 PRINT @ P, "AMOUNT"; : INPUT A$ :
320 IF A$ = "N" THEN 340 ELSE M = VAL(A$)
330 IF M >= 1E6 OR (M = 0 AND A$ <> "0") THEN GOSUB 1000 : GOTO 300
340 PRINT @ P + 6, CHR$(30); TAB(44) USING "$###,###.##-"; M
400 A$ = "" : P = 512
410 PRINT @ P, "PAYEE NUMBER"; : INPUT A$
420 IF A$ = "N" THEN 440 ELSE N = VAL(A$)
430 IF N < 1 OR N > 99 OR N <> INT(N) THEN GOSUB 1000 : GOTO 400 ELSE
    IF P$(N) = "" THEN GOSUB 1000 : GOTO 400
440 PRINT @ P + 5, CHR$(30); " NAME"; TAB(55 - LEN(P$(N))) P$(N)
500 A$ = "" : P = 640
510 PRINT @ P, "ACCOUNT NUMBER"; : INPUT A$
520 IF A$ = "N" THEN 540 ELSE A = VAL(A$)
530 IF A <1 OR A > 99 OR A <> INT(A) THEN GOSUB 1000 : GOTO 500 ELSE IF
    A$(A) = "" THEN GOSUB 1000 : GOTO 500
540 PRINT @ P + 7, CHR$(30); " NAME";TAB(55 - LEN(A$(A))) A$(A)
590 PRINT
600 INPUT "CORRECTIONS (Y OR N)"; A$
610 IF A$="Y" THEN 100
620 IF A$ <> "N" THEN GOSUB 1000 : GOTO 600
630 '''' PROCESSING STEPS FOR EACH TRANSACTION
640 GOTO 100 : '''' ENTER NEXT TRANSACTION
1000 PRINT @ P, CHR$(30); "ERROR -- PLEASE REENTER";
1010 FOR I = 1 TO 500
1020 NEXT
1030 PRINT @ P, CHR$(30)
1100 RETURN
```

Is Your Apple Hung Up by a String?



Bombproof Data Entry

Greg Kielian

Here are two routines for your Apple II that provide a reasonable solution for many of the errors that crop up when programs request string input from users.

These routines allow all alphanumeric characters to be typed in, including commas (no more "?EXTRA IGNORED")! Characters such as 'control C' are filtered out. Each word or string entry has a fixed maximum length which is used to prompt the operator. This fixed length is handy for formatting files, tables, etc. The strings can, if need be, be converted to numbers after entry. Both routines were written in Applesoft Basic.

The first routine inputs data in a conventional echoing mode, with the cursor moving rightword. The second

The second offers an unusual and attractive leftward motion of the input string, with a fixed cursor.

offers an unusual and attractive leftward motion of the input string, with a fixed cursor. Each of these can be used as subroutines, to act as very effective filters and error traps for your user-oriented programs.

Two important variables are:

L — Maximum length of string to be input.

B$ — Where the entered data will reside at the end of the routine.

Forward-Stepping Input (Program 1)

The most important lines of the program are:

200 — Display prompting. Also depends on lines 180, 190, 210, 220 and 330. HTAB values are dependent on the length of the prompt line. In this case line 150 or "LAST NAME:" which is ten characters long and requires a minimum HTAB of 11 (or 10 + I where I is greater than 0).

Greg Kielian, P.O. Box 1084, Longmont, CO 80501.

ASCII CHARACTER CODES (FROM THE APPLESOFT II BASIC PROGRAMMING MANUAL)

DEC = ASCII DECIMAL CODE
HEX = ASCII HEXADECIMAL CODE
CHAR = ASCII CHARACTER NAME
N/A = NOT ACCESSIBLE DIRECTLY FROM THE APPLE II KEYBOARD

DEC	HEX	CHAR	WHAT TO TYPE	DEC	HEX	CHAR	WHAT TO TYPE
Ø	ØØ	NULL	CTRL @	48	3Ø	Ø	Ø
1	Ø1	SOH	CTRL A	49	31	1	1
2	Ø2	STX	CTRL B	5Ø	32	2	2
3	Ø3	ETX	CTRL C	51	33	3	3
4	Ø4	ET	CTRL D	52	34	4	4
5	Ø5	ENQ	CTRL E	53	35	5	5
6	Ø6	ACK	CTRL F	54	36	6	6
7	Ø7	BEL	CTRL G	55	37	7	7
8	Ø8	BS	CTRL H OR ←	56	38	8	8
9	Ø9	HT	CTRL I	57	39	9	9
1Ø	ØA	LF	CTRL J	58	3A	:	:
11	ØB	VT	CTRL K	59	3B	;	;
12	ØC	FF	CTRL L	6Ø	3C	<	<
13	ØD	CR	CTRL M OR RETURN	61	3D	=	=
14	ØE	SO	CTRL N	62	3E	>	>
15	ØF	SI	CTRL O	63	3F	?	?
16	1Ø	DLE	CTRL P	64	4Ø	@	@
17	11	DC1	CTRL Q	65	41	A	A
18	12	DC2	CTRL R	66	42	B	B
19	13	DC3	CTRL S	67	43	C	C
2Ø	14	DC4	CTRL T	68	44	D	D
21	15	NAK	CTRL U OR →	69	45	E	E
22	16	SYN	CTRL V	7Ø	46	F	F
23	17	ETB	CTRL W	71	47	G	G
24	18	CAN	CTRL X	72	48	H	H
25	19	EM	CTRL Y	73	49	I	I
26	1A	SUB	CTRL Z	74	4A	J	J
27	1B	ESCAPE	ESC	75	4B	K	K
28	1C	FS	N/A	76	4C	L	L
29	1D	GS	CTRL SHIFT-M	77	4D	M	M
3Ø	1E	RS	CTRL ↑	78	4E	N	N
31	1F	US	N/A	79	4F	O	O
32	2Ø	SPACE	SPACE	8Ø	5Ø	P	P
33	21	!	!	81	51	Q	Q
34	22	"	"	82	52	R	R
35	23	#	#	83	53	S	S
36	24	$	$	84	54	T	T
37	25	%	%	85	55	U	U
38	26	&	&	86	56	V	V
39	27	'	'	87	57	W	W
4Ø	28	(	(	88	58	X	X
41	29	)	)	89	59	Y	Y
42	2A	*	*	9Ø	5A	Z	Z
43	2B	+	+	91	5B	[	N/A
44	2C	,	,	92	5C	\	N/A
45	2D	-	-	93	5D	]	] (SHIFT-M)
46	2E	.	.	94	5E	↑	↑
47	2F	/	/	95	5F	←	N/A

Table 1.

master charge
THE INTERBANK CARD

"COMPUTERS 'R' US"

VISA

IN CALIFORNIA, BACKORDER OR TECHNICAL INFORMATION:
(714) 698-8088

A CONSUMER COMPUTERS SUBSIDIARY
UNBEATABLE MAIL ORDER DISCOUNTS

TOLL FREE ORDER LINE:
1-800-854-6654

apple computer
Authorized Dealer

APPLE COMPUTER ADD-ONS

DISK II DRIVE ONLY	425
GRAPHICS TABLET	655
SILENTYPE PRINTER w/int. card	515
SSM AIO SERIAL/PARALLEL I/O kit	155
SYMTEC LIGHT PEN card	215
SVA 8 INCH DISK CONTROLLER card	335
VIDEX VIDEOTERM w/graphics ROM	335
DC HAYES MICROMODEM II	319
CORVUS 10 MEGABYTE HARD DISK	4395
ALF MUSIC SYNTHISIZER	235
SPEECHLINK 2000 (64 word vocab)	215
SUP-R—MOD RF TV MODULATOR	30
SUP-R-TERMINAL (80 col. card)	335
MICROSOFT Z-80 SOFTCARD w/CP/M	299
MICROWORKS DS-65 DIGISECTOR	339
PARALLEL PRINTER Int. card	145
COMMUNICATIONS CARD w/conn	185
HI-SPEED SERIAL Int. card	145
PASCAL LANGUAGE SYSTEM	425
CENTRONICS PRINTER Int. card	185
APPLESOFT II FIRMWARE card	149
INTEGER BASIC FIRMWARD card	149
APPLE CLOCK/CALENDAR card	225
SUPERTALKER SD200 SPEECH SYNTH	245
ROMPLUS+(w/kybrd filter)	165
ROMWRITER system	155
INTROL/X-10 BSR SYSTEM	245
MUSIC SYSTEM (16 voices/stereo)	465
A/D + D/A CONVERTER 16 CHANNELS	319
EXPANSION CHASSIS (8 slots)	555

APPLE II PLUS
OR APPLE II STANDARD
16K FOR ONLY
$925
48K $1049
DISK II
W/Controller $485

SOFTWARE

FORTRAN for language system	165
DOS 3.3	49
THE CONTROLLER Gen. Bus	519
THE CASHIER Retail Mgmt. & Inv.	199
APPLEWRITER Word Processor	65
APPLEPOST Mailing List system	45
APPLEPLOT Graph & Plot system	60
VISI-CALC by Personal Software	120
CCA DATA MANAGEMENT Data Base	85
APPLEBUG ASSEMBLER/DISSASSEMBLER	75
APPLE DOS TOOLKIT	65

VIDEO MONITORS

LEEDEX VIDEO 100	129
SANYO 9" B&W	165
SANYO 15" B&W	245
PANACOLOAR 10" COLOR	329
ZENITH/HEATH 13" COLOR	399
NEC 12" HI-RES COLOR	875
NEC 12" LO-RES COLOR	399
NEC 12" GREEN PHOSPHER	239

PRINTERS

ANADEX DP-8000	775
ANADEX DP-9500	1350
CENTRONICS 737	825
PAPER TIGER IDS-440 w/gr	895
TRENDCOM 200	519
EPSON TX-80 w/graphics	729
EPSON MX-80 132 col	620
NEC SPINWRITER	2550

WE ALSO CARRY THESE FINE PRODUCTS
PMC-80
ATARI

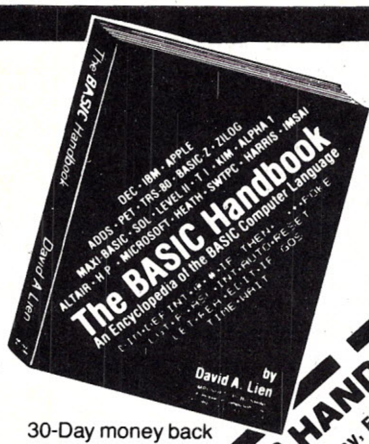
ORDERING INFORMATION: Phone Orders invited using VISA, MASTERCARD, AMERICAN EXPRESS, or bank wire transfers. VISA & MC credit card service charge of 2%, AE credit card service charge of 5%. Mail orders may send charge card number (include expiration date), cashier's check, money order or personal check (allow 10 business days to clear.) Please include a telephone number with all orders. Foreign orders (excluding Military PO's) add 10% for shipping and all funds must be in US dollars. Shipping, handling and insurance in U.S. add 3%. California residents add 6% sales tax. Our low margins prohibit us to send COD or on account. All equipment subject to price change and availability. Equipment is new and complete with manufacturer warranty. We ship most orders within 2 days. Order desk hours are Monday thru Saturday 9-5 PST. Send for FREE 1981 Catalog.

WE ARE A MEMBER OF THE BETTER BUSINESS BUREAU AND THE CHAMBER OF COMMERCE — RETAIL STORE PRICES MAY DIFFER FROM MAIL ORDER PRICES.

PLEASE SEND ORDERS TO: CONSUMER COMPUTERS MAIL ORDER CRU Division 8314 PARKWAY DRIVE, GROSSMONT SHOPPING CENTER NORTH, LA MESA, CALIFORNIA, 92041

CIRCLE 145 ON READER SERVICE CARD

Want to **REALLY UNDERSTAND** The **BASIC** Language?



360 pages
Soft Cover

From the author of the highly acclaimed TRS-80 Users/Learners Manual comes the book you've been asking for! The BASIC Handbook is THE definitive reference and "idea" book, explaining in detail the BASIC language as used in over 50 favorite micros, minis and mainframes.

It's not a dictionary, and not a textbook, but a virtual ENCYCLOPEDIA of the BASIC language. In it is everything you need to know about the most important BASIC statements, functions, operators and commands, explained in a way that you can put them right to work.

30-Day money back Guarantee

This HANDBOOK is written to be used!

With the BASIC Handbook you can finally make those programs found in magazines run on your computer – or know the reason why they can't.

If there is an alternate way to write a program using other BASIC words, the Handbook shows you how. If there is a function needed but your machine doesn't have it, the Handbook gives you a subroutine that accomplishes the same thing. About the only thing it won't help you with is an additional 16K of memory.

Is TRS-80 Level II covered – YES!
Is PET covered – YES!
Is Apple covered – YES!
Sorcerer, Altair, Imsai, Etc.
YES . . . . and over 50 more!

COMPUSOFT™ PUBLISHING
A Division of CompuSoft, Inc.
8643 Navajo Road
San Diego, California 92119

Order Today!

YES, I need the BASIC HANDBOOK

CompuSoft Publishing, 1050 E C-11 Pioneer Way, El Cajon, CA 92020.

Please send ____ copies of The BASIC Handbook. My check for $14.95 each + $1.35 P & H is enclosed. (California add 6% tax)

Name ____
Address ____
City ____ State ____ Zip ____

CC-11-80

I understand my handbook will be shipped promptly and there is a 30-day money back guarantee.

My Computer is a ____

CIRCLE 250 ON READER SERVICE CARD

Entry, cont'd...

230 — GETs character as it's typed.
240 — Checks character to see if it's a carriage return. If it is, control passes to line 370 and the routine is finished.
250 — Checks character to see if it's a back arrow. If it is, lines 260 and 290 remove the last character entered.
300 — If the character has passed the first two tests, the number of characters entered is checked. If it's equal to the maximum number allowed (L), control is sent back to line 210 for a carriage return or back arrow.
310 — If the character is illegal and has an ASCII value less than 32 (refer to Table 1), control is sent back to line 180 and another character is input.
320 — At this point the character has finally passed through the entire filter and is concatenated onto 'B$'.
340 — The character count is incremented and control is sent to fetch another character.
380 — 'B$' is trimmed to the specified length. The program (or subroutine) could be ended at this point, with the desired data in 'B$'.

```
100  HOME
110 M$ = "--------------------"
120 S$ = "                    "
130 L = 10
140  VTAB 10
150  PRINT "LAST NAME:";
160  PRINT  MID$ (M$,1,L)
170 I = 1
180  VTAB 10
190  HTAB (10 + I)
200  PRINT "-"
210  VTAB 10
220  HTAB (10 + I)
230  GET X$
240  IF X$ =  CHR$ (13) THEN 370
250  IF X$ < > CHR$ (8) THEN 300
260  IF I = 1 THEN 180
270 B$ =  MID$ (B$,1, LEN (B$) - 1)
280 I = I - 1
290  GOTO 180
300  IF I = L + 1 THEN 210
310  IF  ASC (X$) < 32 THEN 180
320 B$ = B$ + X$
330  PRINT X$
340 I = I + 1
350  IF I = L + 1 THEN 210
360  GOTO 180
370  CALL  - 868
380 B$ = B$ + S$
390 B$ =  MID$ (B$,1,L)
400  VTAB 15
410  HTAB 1
420  PRINT B$
430  END
```

Routine 1. Forward-Stepping Cursor.

```
100  HOME
110 M$ = "--------------------"
120 S$ = "                    "
130 L = 10
140 B$ =  MID$ (M$,1,L)
150  VTAB 10
160  PRINT "LAST NAME:";
170 I = 1
180  VTAB 10
190  HTAB 11
200  PRINT  MID$ (B$, LEN (B$) - L + 1, LEN (B$))
210  VTAB 10
220  HTAB (11 + L)
230  GET X$
240  IF X$ =  CHR$ (13) THEN 350
250  IF X$ < > CHR$ (8) THEN 300
260  IF I = 1 THEN 180
270 B$ =  MID$ (B$,1, LEN (B$) - 1)
280 I = I - 1
290  GOTO 180
300  IF I = L + 1 THEN 180
310  IF  ASC (X$) < 32 THEN 180
320 B$ = B$ + X$
330 I = I + 1
340  GOTO 180
350 B$ =  MID$ (B$, LEN (B$) - I + 2, LEN (B$))
360 B$ =  MID$ (S$,1,L -  LEN (B$)) + B$
370  VTAB 15
380  HTAB 1
390  PRINT B$
400  END
```

Routine 2. Backward-Stepping String.

Back-Sliding Input (Program 2)

This program operates in much the same way as the first, with the following differences:

140 — 'B$' is initialized with dashes instead of being blank or empty.
200 — Display prompting that resembles backward scrolling. This format also depends on lines 180, 190, 210 and 220. Again HTAB values are dependent on line 160 or "LAST NAME:" which is ten characters long and requires an HTAB of 11.

Examples

Using Routine 1, and entering the name "JOHNSON," gives a display corresponding to the following:
(Note: ∧ indicates cursor position)

Step	Display	
1	LAST NAME: ∧----------	
2	LAST NAME: J ∧---------	
3	LAST NAME: JO ∧--------	
4	LAST NAME: JOH ∧-------	
5	LAST NAME: JOHS ∧------	Note that a mistake was made.
6	LAST NAME: JOH ∧-------	Back arrow was typed.
7	LAST NAME: JOHN ∧------	Mistake was corrected.
8	LAST NAME: JOHNS ∧-----	
9	LAST NAME: JOHNSO ∧----	
10	LAST NAME: JOHNSON ∧---	

At step ten a RETURN was typed and a 'B$' contains "JOHNSON   ." Notice the blanks in the last three character positions.

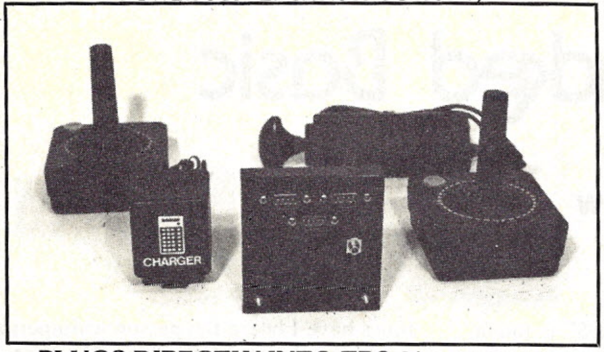
Using Routine 2, and entering the name "STITTSWORTH," results in the following display: (Note: ∧ indicates cursor position)

Step	Display	
1	LAST NAME: ----------∧	
2	LAST NAME: ---------S∧	
3	LAST NAME: --------ST∧	
4	LAST NAME: -------STI∧	
5	LAST NAME: ------STII∧	
6	LAST NAME: -------STI∧	Back arrow was typed.
7	LAST NAME: ------STIT∧	Mistake was corrected.
8	LAST NAME: -----STITT∧	
9	LAST NAME: ----STITTS∧	
10	LAST NAME: ---STITTSW∧	
11	LAST NAME: --STITTSWO∧	
12	LAST NAME: -STITTSWOR∧	
13	LAST NAME: STITTSWORT∧	

At step 13 the only characters recognized are carriage return or back arrow. Since the word didn't fit, it could be modified by retyping it *before* it's actually accepted by the machine. If a RETURN had been typed at step nine, 'B$' would contain "   STITTS." Notice the blanks in the first four character positions. □

*TRS-80 JOYSTICK INTERFACE**

(COMPLETE WITH TWO PROGRAMS, POWER SUPPLY & INSTRUCTIONS)



- **PLUGS DIRECTLY INTO TRS-80 EXPANSION INTERFACE**
- **NO MODIFICATION REQUIRED**
- **ALLOWS YOU TO CONNECT ONE FAIRCHILD™ OR TWO ATARI™ JOYSTICKS**

$65 (PLUS $3.50 SHIPPING)
Joysticks — FAIRCHILD™ OR ATARI™ — $15 EACH
Many other Creative Software products are available for the **PET** and **TRS-80**. If your local dealer doesn't carry Creative Software products or program information, write directly to the address below.

Creative Software

(415) 949-1121
P. O. BOX 4030, MOUNTAIN VIEW, CA 94040
*REQUIRES 16K LEVEL II: JOYSTICKS NOT INCLUDED

SOFTWARE for TRS-80 MODEL I & II

★ General Ledger, C.P.A.
★ Balance-Forward Accounts Receivable
Open-Item Accounts Receivable
★ Contractor Job Costing
★ Accounts Payable
★ Payroll-Union-Piece-Farm-Restaurant
★ Cash Flow Statements
$240.00 each

Quality business software written especially for the Radio Shack computers.
These are not Osborne programs.

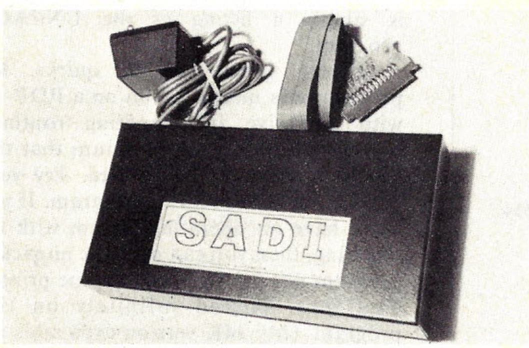
Submit blank diskette for sample report printouts.

COMPUTER SYSTEMS DESIGN, INC.

P.O. Box 735
Yakima, WA 98907
(509) 575-0320

CIRCLE 150 ON READER SERVICE CARD

PET TWO-WAY RS-232 and PARALLEL OUTPUT INTERFACE



SADI - The microprocessor based serial and parallel interface for the Commodore PET. SADI allows you to connect your PET to parallel and serial printers, CRT's, modems, acoustic couplers, hard copy terminals and other computers. The serial and parallel ports are independent allowing the PET to communicate with both peripheral devices simultaneously or one at a time. In addition, the RS-232 device can communicate with the parallel device.

Special Features for the PET interface include:
- Conversion to true ASCII both in and out
- Cursor controls and function characters specially printed
- Transfers programs between PETs over the phone line using a modem.
- Selectable reversal of upper and lower case
- PET IEEE connector for daisy chaining
- Addressable - works with other devices

Special Features for the serial interface include:
- Baud rate selectable from 75 to 9600
- Half or full duplex
- 32 character buffer
- X-ON, X-OFF automatically sent
- Selectable carriage return delay

Special Features for the parallel interface include:
- Data strobe - either polarity
- Device ready - either polarity
- Centronics compatible

Complete with power supply, PET IEEE cable, RS-232 connector, parallel port connector and case. Assembled and tested.

SADIa (110VAC) $295
SADIe (230VAC) $325

Order direct or contact your local computer store.

CONNECTICUT microCOMPUTER, Inc.
150 POCONO ROAD
BROOKFIELD, CONNECTICUT 06804
TEL: (203) 775-9659 TWX: 710-456-0052

VISA AND M/C ACCEPTED - SEND ACCOUNT NUMBER, EXPIRATION DATE AND SIGN ORDER.
ADD $3 PER ORDER FOR SHIPPING & HANDLING - FOREIGN ORDERS ADD 10% FOR AIR POSTAGE.

Loosening Packed Basic

Nick Jackiw

A while ago, one of the popular computing magazines published a program which PACKED a program together. This would save memory and execution time. With a small program, one that had been compacted, the author thought he had a better chance of selling the program to some magazine.

If you can't read it, it's not worth typing in.

I enjoy this program, and use it frequently. However, on the occasions when I submit material to magazines, the PACKER program is not used. My personal experience and the comments from others have told me this fact is true: If you can't read it, it's not worth typing in. I have spent hours laboring over the cramped style of most programmers. Although it's true that "P.X0$" is much shorter (and therefore less space consuming) than "PRINT X0$", you must admit that the latter registers in the mind much more clearly than the first. So many times have I heard the person whimpering over the latest *Creative*, "Is that a semicolon or a colon?"

UNPACKER, I feel, should be used on every program before hard-copy listings are used. Even when I take printout home overnight for pad & pencil debugging, I run UNPACKER.

```
10!WRITTEN BY NICK JACKIW
20X=INT(RND*5)+1:IF X<1 THEN GOTO 20
25INPUT"GUESS THE NUMBER";A0:IF A0=X THEN 30 ELSE GOTO 25
30&"YOU GOT IT!":&"GOOD JOB!":GOTO 20
40END
```

Figure 1

```
00010     !WRITTEN BY NICK JACKIW
00020     X=INT(RND*5)+1
        \ IF X<1 THEN GOTO 20
00025     INPUT"GUESS THE NUMBER";A0
        \ IF A0=X THEN 30 ELSE GOTO 25
00030     PRINT "YOU GOT IT!"
        \ PRINT "GOOD JOB!"
        \ GOTO 20
00040     END
```

Figure 2

```
00010     PRINT "FILE NAME";
        \ INPUT A0$
00020     OPEN A0$ FOR INPUT AS FILE #1%
00030     PRINT "OUT FILE";
        \ INPUT A0$
00035     OPEN A0$ FOR OUTPUT AS FILE #2%
00037     ON ERROR GOTO 300
00040     INPUT LINE#1%,A0$
        \ IF RIGHT(A0$,LEN(A0$)-2%)=CHR$(10%)+CHR$(13%)+CHR$(0%)
          THEN X0%=-1%
        \ A1$=A1$+LEFT(A0$,LEN(A0$)-3%)
        \ GOTO 40
00042     IF X0%=-1% THEN X0%=0%
        \ A0$=A1$+A0$
00043     A0$=CVT$$(A0$,16%)
00045     X0%=INSTR(1%,A0$,"&")
        \ IF X0%=0% THEN GOTO 100
00046     A0$=LEFT(A0$,X0%-1%)+"PRINT "+RIGHT(A0$,X0%+1%)
        \ GOTO 45
00100     X0%=INSTR(1%,A0$,":")
        \ IF X0%=0% THEN GOTO 200
00102     A0$=LEFT(A0$,X0%-1%)+CHR$(10%)+CHR$(13%)+CHR$(0%)+CHR$(9%)+
          "\ "+RIGHT(A0$,X0%+1%)
        \ GOTO 100
00200     X0$=""
        \ FOR I%=1% TO LEN(A0$)
00201     Z$=MID(A0$,I%,1%)
        \ IF INSTR(1%,"0123456789",Z$) = 0% THEN GOTO 205
00202     X0$=X0$+Z$
00203     NEXT I%
00204     PRINT "COMMANDLESS STATMENT KILLED-"+A0$
        \ GOTO 40
00205     IF LEN(X0$)<5% THEN X0$="0"+X0$
        \ GOTO 205
00206     A0$=X0$+CHR$(9%)+" "+RIGHT(A0$,I%)
00210     PRINT #2%,A0$;
        \ GOTO 40
00300     CLOSE #1%
        \ CLOSE #2%
        \ PRINT "UNPACKING COMPLETE."
        \ END
```

Figure 3

There are three listings with this article. The first shows a sample program. In it, the user guesses a number. Figure 2 shows a readable, beautiful program with exactly the same logic! In fact, the program is the same, simply UNPACKED. Figure 3 is simply a listing of the UNPACK program.

Every system has its quirks. My program was meant to run on a PDP-11, with extensive file handling routines. However, it is not the program that this article is about. It is the idea. Try your hand at an UNPACKing program. If you don't have an operating system with file handling, then you can have the unpacked program put out straight to a line printer. You may expand infinitely on this program. (My 14K version even adds the LET verb.)

Guidelines for making an UNPACKer:

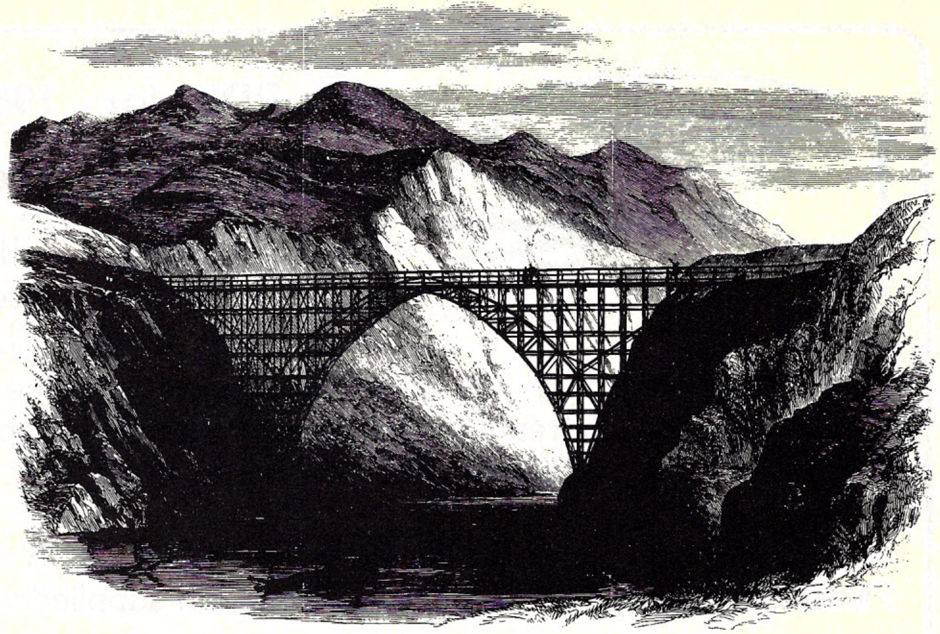
(a) Get rid of all abbreviations.

(b) Add at least three spaces (or a tab) between the line number and the rest of the line.

(c) *Make your results clear!* □

Nick Jackiw, 5806 Murrayhill Place, Pittsburgh, PA 15217.

Shortest Paths



M. L. Shore

The area of mathematics known as Operations Research is probably best known to the layman for such popular applications as PERT (which is based on critical path scheduling). However, a vast number of algorithms have been developed under the umbrella of Operations Research which can be of value to both professional and amateur programmers. One such algorithm is described by Djikstra in his paper entitled "A Note on Two Problems in Connection with Graphs,"

There are many uses for the shortest path algorithm, limited only by our ability to recognize when a problem may be converted into a shortest path in a graph representation.

published in the *Numerische Mathematik,* 1959, Vol. 1, pages 269-271. This algorithm efficiently finds the shortest path from one point in a graph to another point in that graph. Now you need not be put off by terminology such as "shortest path" and "graph" . . . they are merely the official titles for quite simple concepts. Before we describe the algorithm as such, let us first define the necessary terms used later on in this article.

Definitions

We can define a graph, somewhat loosely, as a set of points connected by a set of lines. We can see this diagramatically in Figure 1. Now, connected with each of the lines is a cost. This cost may represent distance (e.g., between two cities), capacity (e.g., of a pipe connecting two waterflows) and so on. We show these costs against the line to which they relate, as in Figure 2.

M. L. Shore, 9 Surrey Flats, Surrey St., Tawa, Wellington, New Zealand.

Figure 1. A graph.

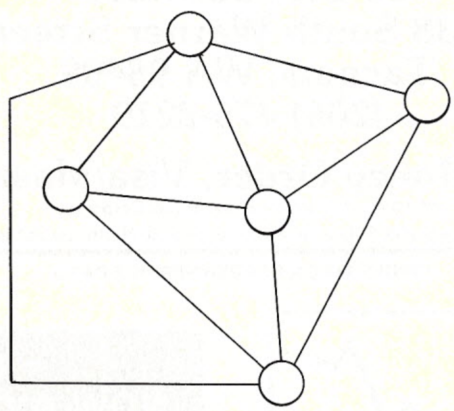
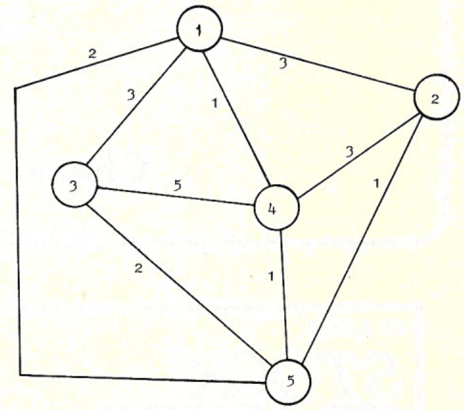


Figure 2. A graph with costs.



However, they can be shown in a different way — as a table of costs as in Table 1. This latter representation is more convenient for the larger graphs, and for those graphs in which the cost of the line from point i to point j is not the same as that of the line from point j to point i (i.e., the graph is not symmetrical, and we call it a directed graph). In this table, each entry (row i, column j, say . . . denoted c_{ij}) represents the cost associated with the line connecting point i to point j. Thus the undirected graph, where $c_{ij}=c_{ji}$, is in fact a special case of the directed graph.

Table 1. The cost table.

FROM \ TO	1	2	3	4	5
1	3	3	3	1	-
2	3	0	-	3	1
3	3	-	0	5	2
4	1	3	5	0	1
5	-	1	2	1	0

Practical Use

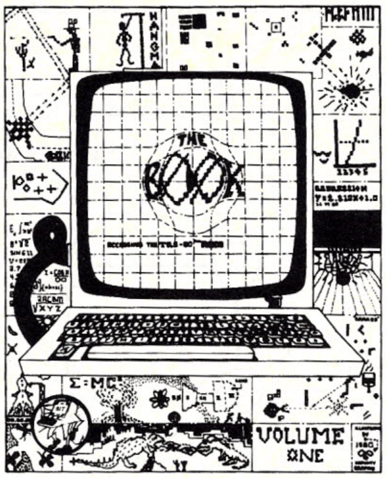
When describing the cost associated with a line, we gave an example of distance as a cost. Let us now look more closely at a problem of a similar nature, where the cost associated with a line denotes the time taken to travel between two points.

Consider the road map as per Figure 3. We have 11 junctions connected by roads, and in brackets we have given the time taken to negotiate that road — i.e., to pass from one junction to the other. This map can easily be converted to a graph, as we can see from Figure 4, by replacing the junctions with points and the roads connecting the junctions with lines connecting the points. Now suppose that we wish to get from junction 2 (J2) to junction 10 (J10) as quickly as possible. This problem can be stated as finding the shortest path between point 2 and point 10 on our graph of Figure 4. Using the Djikstra algorithm we can generate the shortest path as in Figure 5, and by decoding points and lines back to junctions and roads, we see that our best route would be J2 - J7 - J11 - J10, which we would expect to take 7 minutes.

BACK TO BASIC

THE BOOK

ACCESSING THE TRS-80* ROM



If you ever do Assembly language programming, or you just want to know more about your TRS-80 ROM, "THE BØØK" is for you.

Volume I will give you access to over fifty machine language subroutines in the Radio Shack Level II BASIC. It includes information on the numeric data formats and a commented listing of the ROM routines.

"THE BØØK, Volume I", encompasses all arithmetic functions and mathematical operations. There are separate routines for integers, single precision, and double precision numbers and the data format for each of these number types is explained. The routines that perform ASCII to binary and binary to ASCII conversion are identified and explained to provide you a means of data I/O.

A fully commented listing provides the details on the step-by-step execution of these ROM routines. Although a complete disassembly is not provided in order to avoid copyright infringement, you can obtain a complete disassembly using the disassembler program listed in "THE BØØK." Volume I also includes a complete, detailed memory map of the entire machine and a symbol table noting over 500 addresses.

"THE BØØK" will save you hour upon hour of assembler program development time. Don't start programming without it.

Order your copy of "THE BØØK", today!

DEALER INQUIRIES INVITED

Insiders Software Consultants, Inc.
P.O. Box 2441, Dept. S 1
Springfield, VA 22152

*TRS-80 is a trademark of Tandy Corp.

☐ **Please send me Volume I of THE BØØK at $14.95 plus $1.50 for postage.**

NAME: ______

ADDRESS: ______

CITY, STATE ______ & ZIP CODE: ______

☐ Check payable to Insiders Software Consultants, Inc.
☐ MASTER CHARGE MC Bank Code:
☐ VISA Exp. Date: Card Number:

Signature: ______

CIRCLE 205 ON READER SERVICE CARD

The ATARI® Tutorial COMPUTER Calligraphy?

Well, not really! But with the FONTEDIT program in **IRIDIS #2** you can design your own character sets for the **ATARI**. For example, you can create a Russian alphabet, or APL characters, or even special-purpose graphics symbols. These special *fonts* can be saved on disk or tape for later use by your own Basic programs. FONTEDIT is a friendly, easy-to-use program: just grab a joystick and start designing. With our KNOTWORK program, you can design patterns of *Celtic interlace*, (a technique used by 7th century Irish monks to illuminate manuscripts). After you have produced a pretty pattern on the screen of your **ATARI**, you can save it on disk or tape.

Best of all, **IRIDIS #2** comes with a **48-page** *User's Guide*, which gives clear instructions on how to use the programs. The *guide* also provides detailed, line-by-line descriptions of how the programs work. **IRIDIS** programs are written to be studied as well as used.) *Hacker's Delight* presents useful explanations of many of the important PEEK and POKE locations in your **ATARI.**

The **IRIDIS #2** tutorial for the **ATARI** is available now from your local **ATARI** dealer. You get the *User's Guide* and a cassette (or disk) with FONTEDIT, KNOTWORK, and a routine to allow your program to load a custom font.

ATARI is a trademark of **ATARI, Inc.**

To: **The Code Works,** Box 550, Goleta, CA 93017
Please send me **IRIDIS #2** for my **ATARI** immediately.
☐ $15.95 Cassette (needs 16K) ☐ $18.95 Disk (needs 24K)

Name ______

Address ______

City/State/Zip ______

☐ VISA Card Number ______

☐ MasterCharge Expires ______

Phone: (805) 967-0905

Programs for your **ATARI®**

CIRCLE 209 ON READER SERVICE CARD

What you 'C' is what you get!

C Compiler for CP/M*

New, and available now! An easily affordable compiler incorporating most of the features of the full C language.

BD SOFTWARE

System requirements: CP/M and at least 24K of RAM

Variable Types: char, int, unsigned

Composite Types: arrays, structures, unions

Pointers: to variables, structures, unions and functions

Features: is a structured language, all functions (Programs) recursive; more powerful expression operators than any other von Neumann type language; allows free-formatted source; close enough to UNIX** C to make conversions feasible.

Speed: On 2 MHz 8080, the statement for (i = 1; i < 30000; i++) x = 5; takes about 4 seconds to execute.

Package contains: compiler, linker, library manager; standard function library; sample source files include games, a terminal emulator with disk I/O plus the source for many standard library functions; BDS C User's Guide; Book—*The C Programming Language* by Dennis Ritchie and Brian Kernighan of Bell Labs.

Price: **$145**
Manual Alone $25

Recipient of the Computer Lib Seal of Approval

*CP/M is a trademark of Digital Research Corp.
**UNIX is a trademark of Bell Laboratories

Prices reflect distribution on 8" single density diskettes. If a format is requested which requires additional diskettes, a surcharge of $8. per additional diskette will be added

Lifeboat Associates
1651 3rd Ave., New York, NY 10028 (212) 860-0300 Telex: 220501

```
READY.

 10 CLR:PRINT "♥"
 20 PRINT "▒▒▒▒▒▒"
 30 PRINT "           S H O R T E S T   P A T H":PRINT
 35 PRINT "                   (DIRECTIONAL)":PRINT
 40 FOR I=1 TO 2000:NEXT
 50 PRINT "♥":PRINT "▒▒▒"
 60 PRINT "THIS PROGRAM ILLUSTRATES THE USE OF"
 70 PRINT "THF SHORTEST PATH ALGORITHM AS PER"
 80 PRINT "DJIKSTRA (1959)."
 90 PRINT "THE NEED TO FIND A SHORTEST PATH "
 100 PRINT "FROM ONE POINT TO ANOTHER IN A GRAPH"
 110 PRINT "(OR NETWORK) OCCURS IN A VARIETY OF"
 120 PRINT "OPERATIONS RESEARCH APPLICATIONS"
 130 PRINT:PRINT "THE FOLLOWING PROGRAM MERELY"
 140 PRINT "REQUESTS INPUT OF A GRAPH, AND"
 150 PRINT "PRODUCES THE SHORTEST PATH,AS A WAY"
 160 PRINT "OF DEMONSTRATING THE TECHNIQUE"
 170 GET A$:IF A$="" THEN 170
 200 PRINT "♥":PRINT:PRINT
 210 INPUT "HOW MANY POINTS IN THE GRAPH";N
 220 REM
 230 REM ****************************
 240 REM *                          *
 250 REM *     INPUT THE GRAPH      *
 260 REM *                          *
 270 REM ****************************
 280 DIM D(N,N),A(2,N)
 290 FOR I=1 TO N
 300 FOR J=1 TO N
 305 IF I=J THEN 340
 310 PRINT "♥":PRINT:PRINT
 320 PRINT "COST OF LINE";I;"TO";J;:INPUT D(I,J)
 340 NEXT:NEXT
 350 REM
 360 REM ******************************
 370 REM *                            *
 380 REM *    GET THE TWO POINTS      *
 390 REM *                            *
 400 REM ******************************
 410 PRINT "♥":PRINT:PRINT
 420 INPUT "ENTER THE TWO POINT NUMBERS";X1,X2
 510 REM
 520 REM ******************************
 530 REM *                            *
 540 REM *      NOW ENTER THE S/R     *
 550 REM *                            *
 560 REM ******************************
 570 REM
 580 GOSUB 1000
 590 REM
 600 REM ******************************
 610 REM *                            *
 620 REM *  NOW DISPLAY THE RESULTS   *
 630 REM *                            *
 640 REM ******************************
 650 PRINT "♥":PRINT
 660 PRINT "THE SHORTEST PATH IS OF COST";A(1,X1)
 670 PRINT
 680 PRINT "PATH LINES":PRINT "==========":PRINT
 810 I=X1
 820 PRINT I;A(2,I)
 830 IF A(2,I)<>X2 THEN I=A(2,I):GOTO 820
 840 PRINT
 850 STOP
 940 REM ******************************
 950 REM *                            *
 960 REM *  THIS S/R CALCULATES       *
 970 REM * SHORTEST PATH FOR X1-X2    *
 980 REM *                            *
 990 REM ******************************
 1000 FOR I=1 TO N:A(1,I)=D(I,X2):A(2,I)=X2:NEXT
 1010 A(2,X2)=-X2
 1020 I1=X2
 1030 MN=1000
 1040 FOR I=1 TO N
 1050 IF A(2,I)<0 THEN 1080
 1060 IF D(I,I1)+A(1,I1)<A(1,I) THEN A(1,I)=D(I,I1)+A(1,I1):A(2,I)=I1
 1070 IF A(1,I)<MN THEN I2=I:MN=A(1,I)
 1080 NEXT
 1100 I1=I2:A(2,I1)=-A(2,I1)
 1110 IF I1<>X1 THEN 1030
 1200 FOR I=1 TO N:A(2,I)=ABS(A(2,I)):NEXT
 1210 RETURN
READY.
```

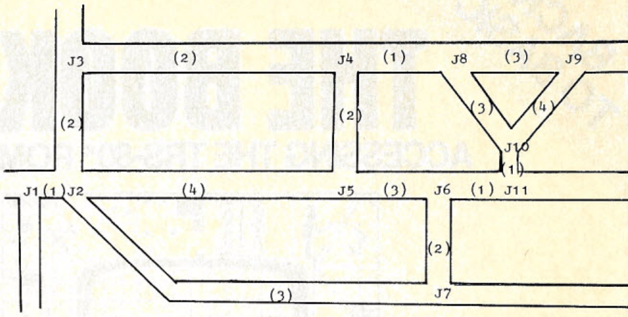


Figure 3. A road map.

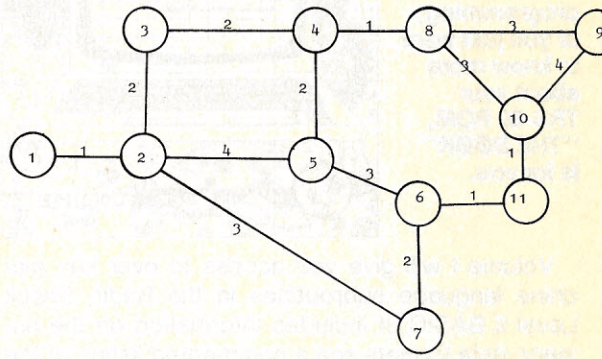


Figure 4. A graphical representation of the road map.

Although in this case we have a small graph, and consequently a small number of possible routes to look at, consider the case where there are, say, 100 junctions and a subsequently large number of possible routes; could we then tell the quickest route by mere examination of the map by eye? I think not; certainly not easily.

The Algorithm

Now to the crux of the article . . . the algorithm. The formal definition of the algorithm by pseudocode can be generated with ease from the demonstration program, lines 1000 onward. Hence, here we present only an explanation of the algorithm.

Initially, we give the to-point specified (say x2) the "permanent" value zero, and give all other points the "tentative" value of the direct distance from that point to point x2. Then, one by one, we compare each tentative point's associated distance with the sum of the distance from the last point to be set as permanent to x2, and the direct distance from this tentative point to the permanent point. The smaller of these two distances then becomes the tentative point's new associated distance.

Next, determine the smallest associated distance for all points still tentative, and declare the point with that smallest distance to be permanent. Then go back to the previous stage, and compare the associated distances of all tentative points with the sum of the distance from the last point to be set as permanent . . . etc.

The algorithm terminates when the

QUALITY PROFESSIONAL & BUSINESS SOFTWARE
FOR THE APPLE II AND APPLE PLUS COMPUTERS

LETTER MASTER - This is a lightning fast Word Processor which is simple to use and operates like a character oriented system. It features a menu driven operating system which allows entry, storage, editing, draft printing, and final justified printing. Editing includes line corrections (add up to 200 characters if you want), global search, string replacement, delete, add, and merge. Text floats as needed to make up lines of text. Print justified, unjustified, or a mixture. System prepares and prints form letters with its own 500 name/diskette full feature mailing list program. Applesoft II Disc. Item M-34, $139.95. Processor without mailing list and form letter, Item M-33, $69.95.

PROGRAMMER PLUS - A 16 lesson course in Applesoft II on two full disks or super-load tape which will teach you all you need to know to program in BASIC. Lessons cover all string, math and logical operations necessary for personal, scientific or business applications. Special units teach graphics and sound to add a little 'Apple Class' to your programs. Applesoft II, Item 35. Tape, $24.95. Disc, $34.95.

SCIENTIST - A powerful scientific data base and statistics program which will turn your Apple into a mathematical and graphics tool. It will do simple statistics or such complicated functions as Chi- Square, Normal, Student-t, and Poisson. With the Data Base element you can build your own analytical programs. Applesoft II Disc, $89.95.

Shipping: $3.00 order. CA residents add sales tax

TO ORDER CALL TOLL FREE 24 HOURS
800-854-0561 Ext. 802
In CA 800-432-7257
OR WRITE FOR TECHNICAL ASSISTANCE TO
MONUMENT COMPUTER SERVICE
Village Data Center - P.O. Box 603
Joshua Tree, California 92252

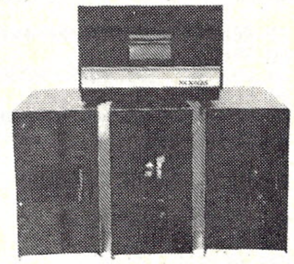
CIRCLE 152 ON READER SERVICE CARD

Being a deaf dancer is really something.

The President's Committee on Employment of the Handicapped Washington, D.C. 20210
The School of Visual Arts Public Advertising System

SAVE ON ADD-ON PRODUCTS FOR TRS-80

The largest family of disk drives from the largest supplier, drives come complete with power supply and cabinet.



MTI-40 Disk Drive, 35 & 40 track	**$369**
TF-1 Perfec FD200, 40 track, use both sides	$389
TF-3 Shugart SA400, 35 track, same as tandy	$389
TF-5 MPI B51, 40 track	$379
TF-70 Micropolis, 77 track with 195K of storage	$639
TDH-1 Dual sided drive, 35 track	$499
Maxi Disk 2: 10 Megabyte (fixed) Winchester Technology	**$5349**

NEW PRODUCTS

16K Memory	$86
Modem	$179
Expansion Interface 32K	$499
AC Isolator	$47.95

PRINTERS

DP800 Anadex, 80 column, 112cps	$949
LP779 Centronics 779	$1099
LP730 Centronics 730	$950
LP700 Centronics 700	$1395
LP701 Centronics 701	$1759
LP702 Centronics 702	$1995
SPW-1 Spinwriter-NEC	$2525

NEW! LINE PRINTER BASE 2

Base 2 Printer 80, 132 col., graphics
60 LPM with tractors **$599**

*** DRIVES FOR ANY MICROCOMPUTER ***
Does not include power supply & cabinet.

MOD II DISK DRIVES NOW AVAILABLE

Perfec FD200	$282
Perfec FD250 (dual head)	$399
Shugart SA400 (unused)	$286
Shugart SA800	$479
MPI B52 .... **$349** B51	$279

SOFTWARE

Disk Drive Motor Speed Test	$19.95
New DOS+ with over 200 modifications and corrections to TRS-DOS	$99
New DOS+ 40 track	$110
AJA Word Processor	$75
AJA Business Program	$250
Racet Infinite Basic	$49.95
Disk Drive Alignment Program	$109
Radix Data Base Program	$99.95
Electric Pencil	$150

ALL PRICES CASH DISCOUNTED. FREIGHT FOB/FACTORY

MICROCOMPUTER TECHNOLOGY INCORPORATED
3304 W. MacArthur
Santa Ana, CA 92704
(714) 979-9923

Apparat, Inc.
7310 E. Princeton Ave.
Denver, CO 80222
(303) 758-7275

CIRCLE 151 ON READER SERVICE CARD

See us at the west Coast Computer Faire, Booth #1109C

Paths, cont'd...

```
READY.

 10 REM *************************
 20 REM *                       *
 30 REM *   EXAMPLE PROGRAM 1   *
 40 REM *                       *
 50 REM *************************
 60 CLR:PRINT "♥"
 80 DIM B$(1000)
 90 PRINT:INPUT "STRING";A$
 100 TS=TI
 110 FOR I=1 TO 1000
 120 B$(I) = A$
 130 NEXTI
 140 TC=TI-TS
 150 PRINT "TIME TAKEN WAS";TC
 160 STOP
READY.
```

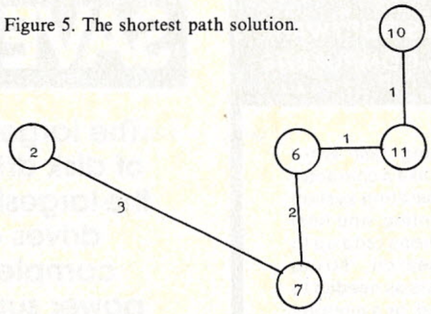
```
READY.

 10 REM *************************
 20 REM *                       *
 30 REM *   EXAMPLE PROGRAM 2   *
 40 REM *                       *
 50 REM *************************
 60 CLR:PRINT "♥"
 80 DIM B$(1000)
 90 PRINT:INPUT "STRING";A$
 100 TS=TI
 110 FOR I=1 TO 1000
 120 B$(I)=A$
 130 NEXTI
 140 TC=TI-TS
 150 PRINT "TIME TAKEN WAS";TC
 160 STOP
READY.
```

```
READY.

 10 REM *************************
 20 REM *                       *
 30 REM *   EXAMPLE PROGRAM 3   *
 40 REM *                       *
 50 REM *************************
 60 CLR:PRINT "♥"
 70 I=0:A$=""
 80 DIM B$(1000)
 90 PRINT:INPUT "STRING";A$
 100 TS=TI
 110 FOR I=1 TO 1000
 120 B$(I)=A$
 130 NEXTI
 140 TC=TI-TS
 150 PRINT "TIME TAKEN WAS";TC
 160 STOP
READY.
```

number of points	time taken (elapsed secs)
5	1.0
8	2.3
10	3.5
15	7.3
20	12.2
30	26.2
50	68.9

Table 2. Elapsed time in seconds on a PET 2001.

Figure 5. The shortest path solution.




```
READY.

 10 REM *************************
 20 REM *                       *
 30 REM *   EXAMPLE PROGRAM 4   *
 40 REM *                       *
 50 REM *************************
 60 CLR:PRINT "♥"
 70 I=0:A$=""
 80 DIM B$(1000)
 90 PRINT:INPUT "STRING";A$
 100 TS=TI
 110 FOR I=1 TO 1000
 120 B$(I)=A$
 130 NEXT
 140 TC=TI-TS
 150 PRINT "TIME TAKEN WAS";TC
 160 STOP
READY.
```

```
READY.

 10 REM *************************
 20 REM *                       *
 30 REM *   EXAMPLE PROGRAM 5   *
 40 REM *                       *
 50 REM *************************
 60 CLR:PRINT "♥"
 70 I=0:A$=""
 80 DIM B$(1000)
 90 PRINT:INPUT "STRING";A$
 100 TS=TI
 110 FOR I=1 TO 1000:B$(I)=A$:NEXT
 140 TC=TI-TS
 150 PRINT "TIME TAKEN WAS";TC
 160 STOP
READY.
```

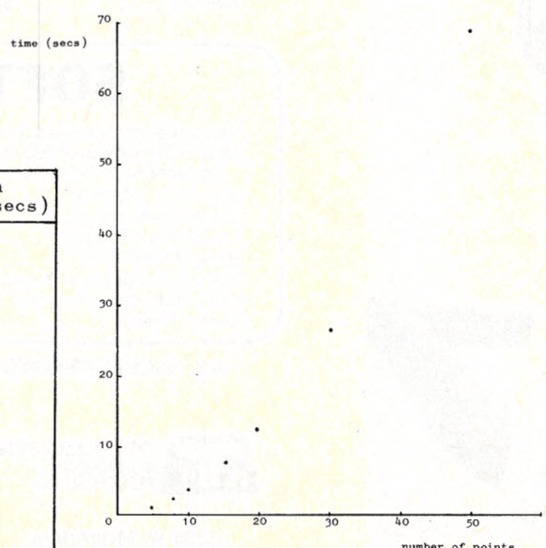


Figure 6.

Time taken to solve problems on a Commodore PET.

from-point (say x1) has been set as permanent. Then the shortest path from x1 to x2 is of that cost associated with point x1 (its associated distance). The actual path itself can be determined by keeping track of the points to which all points connect on their shortest path to the point x2.

The Implementation

In the implementation as shown in the demonstration program, the program reads a directional graph. (Lines 290-340 could be easily modified to input a nondirectional graph, and hence speed up graph input by a factor of two. This, however, makes no difference to the algorithm.) The graph is stored in a nxn array D, where n is the size of (the number of points in) the graph, and the associated distances for the permanent and tentative points are stored in the part array a(1,). The point to which the permanent/tentative point is connected on its shortest path to x2 is stored in the part array a(2,). The distinction between permanent and tentative status is given by the sign of this latter part array, such that negative a(2,) entries for a point denote it as being permanent.

Consider the case where there are, say, 100 junctions and a subsequently large number of possible routes; could we then tell the quickest route by mere examination of the map by eye? I think not; certainly not easily.

Timings

The time taken for the algorithm to process the graph obviously depends upon how soon point x1 is declared permanent. However, we ran a number of problems through the demonstration program such that point x1 was the last point to be declared permanent, and the results are shown in tabular form, Table 2, and graphical form, Figure 6. It can be seen from the timings, and calculated from the algorithm, that the time taken to solve a particular problem will be related to the number of points squared.

Conclusions

Thus we have an efficient algorithm for calculating shortest paths. The demonstration program does not pretend to be the fastest implementation of the algorithm (you may be interested in seeing just how fast you can make the algorithm run on your computer), but it is a compromise between speed and readability.

Figure 7. A wargaming situation.

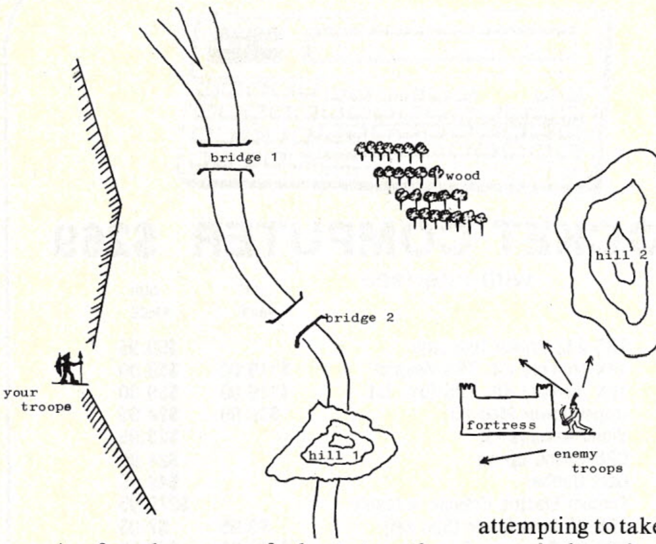
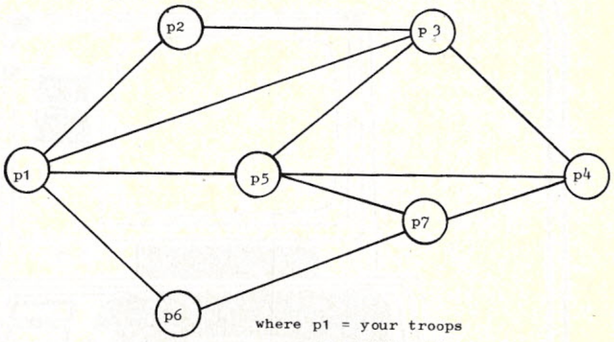


Figure 8.
A graph representation of the wargaming situation.

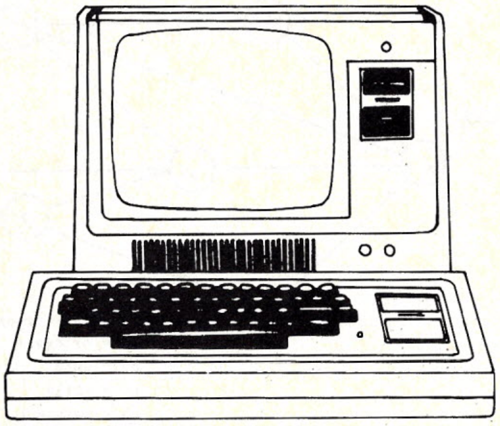


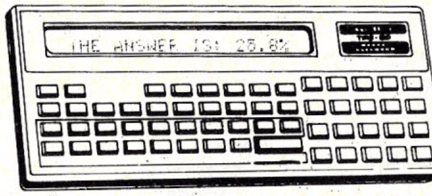

```
where p1 = your troops
      p2 = bridge 1
      p3 = wood
      p4 = hill2
      p5 = bridge 2
      p6 = hill 1
      p7 = fortress
```

As for the uses of shortest path techniques, there are many. The technique is used to a great extent as part of a solution to other problems in optimization, but does form a solution in itself for problems like the quickest route. Another stand-alone use of the technique could be as follows.

Consider the situation depicted in Figure 7 of a typical wargame problem. Your troops must take the enemy fortress, and in order to do so have to move across enemy occupied territory. In the process of attempting to take the fortress, your troops may need to take one or more intermediate objectives, such as bridges, hills and woods. By deciding where you think the enemy is, and in what concentration, you will be able to calculate the cost of taking one objective from a previously taken objective. This cost may be expressed in terms of casualties expected in taking the objective, or probability of failure, for example. Then by converting this problem into a shortest path problem, you can find the shortest path between point 1 and point 7, as in Figure 8. This then provides you with the best plan of attack for your troops. Of course as the game proceeds, you may want to continually reevaluate the situation by updating your probability or casualty cost table in the light of new experience, and producing a new shortest path solution.

Thus there are many uses for the shortest path algorithm, limited only by our ability to recognize when a problem may in fact be converted into a shortest path in a graph representation! So here is an efficient algorithm; it is now up to you to go forth and apply it. □



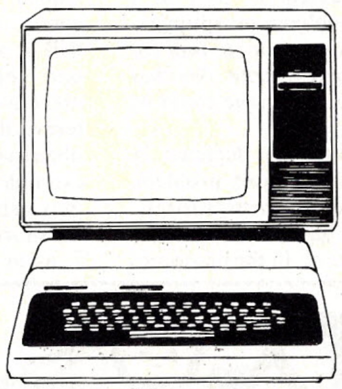


POCKET COMPUTER $269
with interface

	LIST PRICE	OUR PRICE
TRS-232 Printer Interface		$59.95
16K Memory Kit, TRS-Keypad	$119.00	$59.00
16K Memory Kit, TRS-Exp. Int.	$119.00	$59.00
Upper/lower Mod Kit	$59.00	$24.95
Video Reverse Kit		$23.95
CPU Speed-up Kit		$24.95
Data Dubber		$49.95
Percom Electric Crayon, w/cable		$279.95
TRS-80 Dust Cover (3pc set)	$9.95	$7.95
TRS-80 Computer Case	$109.00	$99.95
TRS-80 Monitor Case	$84.00	$84.00

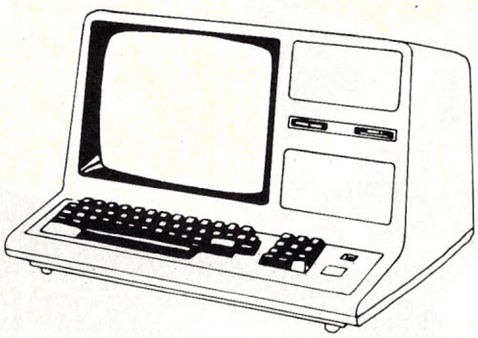
MODEL I $699

	LIST PRICE	OUR PRICE
Model-I, Level-II, 4K	$649.00	$619.00
Model-I, 16K no Keypad	$768.00	$669.00
Model-I, 16K w/Keypad	$849.00	$729.00
Model-II, 64K RAM	$3899.00	$3799.00
Model-III, 16K RAM	$999.00	$929.00
Model-III, 32K Dual Disk	$2495.00	$2299.00
Pocket Computer w/Interface	$298.95	$269.00
TRS-80 Color Computer	$399.00	$359.00
TRS-80 Color Computer Expanded	$599.00	$519.00
COMM-80 Interface	$179.95	$159.95
CHATTER BOX Interface		$259.95
DISK-80 Interface	$349.95	$329.95
Expansion Interface, no RAM	$299.00	$279.00
Expansion Interface, 16K RAM	$418.00	$339.00
Expansion Interface, 32K RAM	$537.00	$399.00
RS-232-C Board	$99.00	$89.00



COLOR COMPUTER $359

	LIST PRICE	OUR PRICE
Percom, TFD-100, 40-track	$429.95	$399.00
Percom, Dual TFD-100 Drives	$849.00	$799.00
Percom, TFD-40, 40-track	$399.95	$379.00
Percom, TFD-200, 77-track	$675.00	$629.00
Hardside, 40-track Disk Drive	$399.00	$359.00
Percom Data Separator		$29.95
Percom Extender Card	$15.95	$15.00
2-Drive Cable	$29.95	$29.00
4-Drive Cable	$39.95	$39.00



MODEL III $929

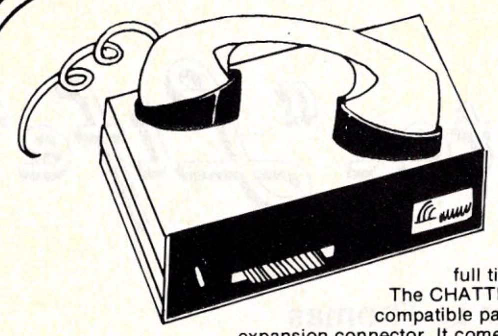
TERMS:

TERMS: Prices and specifications are subject to change. HARDSIDE accepts VISA & MASTERCARD. Certified checks and Money Orders; Personal checks accepted (takes 3 weeks to clear). HARDSIDE pays all shipping charges (within the 48 states) on all PREPAID orders OVER $100.00. On all orders over $100 a $2.50 handling charge must be added. COD orders accepted (orders over $250 require 25% deposit) there is a $5.00 handling charge. UPS,Blue Label, and Air Freight available at extra cost.

TO ORDER TOLL-FREE:
1-800-258-1790
(In NH call 673-5144)

MasterCard VISA

HARDSIDE™



The Chatterbox

A TRS-80 Interfacing Alternative

The CHATTERBOX is a unique packaging combination of the presently available COMM-80 I/O Interface for the TRS-80* and an acoustic modem. This one box is all that is required to turn even a barebones 4K TRS-80* into a full time–sharing terminal.

The CHATTERBOX includes built-in programmable 50-19200 baud serial port, a Centronics compatible parallel printer port, a 300 baud acoustic originate modem, and a spare TRS-BUS expansion connector. It comes complete with power supply, ribbon cable and connector, user's manual, and smart terminal software for immediate operation. When the modem is in use, the complete data conversion is automatically routed to the serial output port where it can be logged on a printer.

The CHATTERBOX is the only peripheral needed to allow a TRS-80* to communicate with time–sharing systems such as MICRONET and the SOURCE.

It is completely hardware and software compatible with existing TRS-80* products and connects either to the keyboard connector or screen printer port on the RS Expansion Interface. Features: Full 8-bit parallel port; RS-232-C serial port (up to 19,200 baud); Acoustic modem; TRS-BUS connector for future expansion; Connects to Keyboard or E.I.; Includes terminal software; Users manual; Power supply. $259.95

PRINTERS

	LIST PRICE	OUR PRICE
Centronics 730	$795.00	$749.00
Centronics 730-3	$895.00	795.00
Centronics 737	$995.00	$869.00
Centronics 779	$1395.00	$1095.00
Centronics 779 w/lower case	$1595.00	$1195.00
NEC 5510 SpinWriter	$3195.00	$2595.00
NEC 5520 SpinWriter	$3395.00	$2895.00
NEC 5530 SpinWriter	$3195.00	$2495.00
NEC Tractor-Feed Option	$249.00	$225.00
LRC 7000$ (64-col.)	$405.00	$299.00
LRC 7000$ (40-col.)	$389.00	$289.00
Okidata Microline-80	$800.00	$699.00
Tractor-Feed Option	$140.00	$129.00
RS-232-C Option	$299.00	$279.00

LRC to TRS-80	$20.00
LRC to PET, IEEE	$59.00
LRC to RS232C, male or female	$65.00
730 or 737 to TRS-80	$29.00
NEC or 779 to TRS-80	$35.00
RS-232-C to RS-232-C, male to male	$24.95

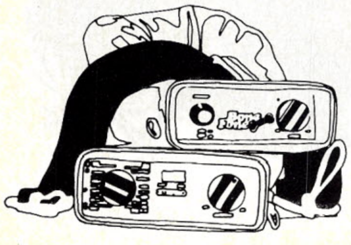
GENERAL INTEREST

	LIST PRICE	OUR PRICE
BSR System X-10 Home Controller	$124.95	$109.95
BUSY BOX, TRS-80	$114.95	$99.95
BUSY BOX, S-100	$119.95	$114.95
AC-SFK-31 Line Filter	$24.95	$19.95
ISO-2 Line Filter & Isolator	$56.95	$49.95
ISO-2/CBS Line Filter-Isolator	$70.95	$59.95
ISO-7/CB Super Filter-Isolator	$146.95	$99.95
Mini-Flex Diskette File	$24.95	$19.95
CASIO C-80 Calculator Watch	$49.95	$44.95
BONE FONE	$69.95	$56.95
LOGOS-9 Printing Calculator	$99.95	$79.95

SPECIAL

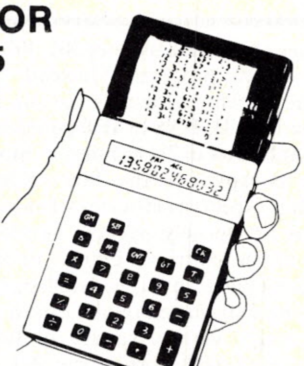
CASIO C-80 CALCULATOR WATCH $44.95

(plus $2.50 postage and handling)



BONE FONE $56.95

(plus $2.50 postage and handling)



OLIVETTI LOGOS-9 PRINTING CALCULATOR $79.95

(plus $2.50 postage and handling)

TO ORDER TOLL-FREE
1-800-258-1790
(In NH call 673-5144)

HARDSIDE™

MasterCard VISA

CIRCLE 183 ON READER SERVICE CARD

A Faster "Life"

Arthur L. Thomas

Creative Computing's *Basic Computer Games* includes Baker and North's ingenious program for Conway's population dynamics simulation, Life. Unfortunately, on some machines this program is a bit sluggish. For instance, on my Wang PCS II it takes about 15½ minutes for it to generate the first 21 tableaux of Baker and North's sample run.

Much of this time is spent repeatedly executing program steps 530-570. The simulation uses a matrix, and the program identifies its "active" portion — a rectangle within which populations can change during the short run. For each cell in that rectangle, steps 530-570 obtain population information about both it and its eight neighboring cells. Thus, each cell that lies in the rectangle's interior will be investigated 9 times per tableau. During the first 21 tableaux, there are 25,992 such investigations!

The remedy is to change the way that one represents populations. The program FASTLIFE uses the same version of Basic, and makes the same assumptions, as the Baker-North program. But each cell of its primary (Z) matrix reflects both whether it and whether its neighbors are occupied. This allows the program to skip over all cells except those in which a birth or death is about to occur. FASTLIFE also saves a little more time by not doing the testing for illegal values performed by lines 301-307 of the Baker-North program. (Instead, an illegal value merely terminates the run.)

Its primary matrix reflects both whether it and its neighbors are occupied. This allows the program to skip over all cells except those in which a birth or death is about to occur.

On the PCS II, FASTLIFE generates the first 21 tableaux in 7¼ minutes — more than twice as rapidly as the Baker-North program. It is especially efficient whenever populations are loosely packed, as when the center of the matrix is empty, or the population includes small, isolated stable groups. As an extreme example, for an 18 by 18 rectangle with a square of four occupied cells in each corner, the Baker-North program requires 101 seconds per iteration, while FASTLIFE needs only 21½.

Finally, if your Basic includes a few elementary matrix operations, you can make FASTLIFE run even faster. Interested readers are invited to request from me a PCS II program that generates the first 21 tableaux in less than five minutes, and the 18 by 18 rectangle in 13½ seconds.

Arthur L. Thomas, Rice University, P.O. Box 1892, Houston, TX 77001.

```
10 REM "FASTLIFE"
20 PRINT TAB(26); "J.H. CONWAY'S 'LIFE'": PRINT
30 REM PROGRAM BY ARTHUR L. THOMAS, THE UNIVERSITY OF KANSAS, THE
40 REM          SCHOOL OF BUSINESS, LAWRENCE, KANSAS  66045
50 PRINT TAB(6); "BEGIN EACH INPUT LINE WITH A DOT, THEN ASTERISKS AND
   SPACES."
60 PRINT TAB(10); "TO END INPUT, RESPOND TO THE PROMPT WITH / (RETURN)."
   :PRINT :PRINT
70 DIM Y(26,26), Z(26,26), A$(21)21
80 L = L+1
90 INPUT A$(L)
100 IF LEFT$(A$(L),1) = "/" THEN 150
110 IF W >= LEN(A$(L))-1 THEN 130
120 W = LEN(A$(L))-1
130 IF L = 20 THEN 150
140 GOTO 80
150 L = L-1: L1 = INT((26-L)/2)+1: W1 = INT(((26-W)/2)+1.5)
160 L9 = L+L1-1: W9 = W+W1-1
170 FOR I = L1 to L9
180 FOR J = W1 TO W9
190 IF MID$(A$(I-L1+1),(J-W1+2),1) = " " THEN 260
200 Z(I,J) = Z(I,J)+10
210 FOR I1 = -1 TO 1
220 FOR J1 = -1 TO 1
230 Z(I+I1,J+J1) = Z(I+I1,J+J1)+1
240 NEXT J1
250 NEXT I1
260 NEXT J
270 NEXT I
280 L2 = L1: L8 = L9: W2 = W1: W8 = W9: L1,L9 = INT((L1+L9)/2): W1,W9 =
    INT((W1+W9)/2): Q = 0
290 FOR I = 1 TO L2-2 :PRINT :NEXT I
300 FOR I = L2-1 TO L8+1
310 FOR J = W2-1 TO W8+1
320 IF Z(I,J) > 9 THEN 480
330 IF Z(I,J) <> 3 THEN 560
340 Y(I,J) = Y(I,J)+10
350 FOR I1 = -1 TO 1
360 FOR J1 = -1 TO 1
370 Y(I+I1,J+J1) = Y(I+I1,J+J1)+1
380 NEXT J1
390 NEXT I1
400 IF L1 < I THEN 420
410 L1 = I
420 IF L9 > I THEN 440
430 L9 = I
440 IF W1 < J THEN 460
450 W1 = J
460 IF W9 > J THEN 560
470 W9 = J: GOTO 560
480 Q = Q+1: PRINT TAB(J+22); "*";
490 IF ABS(Z(I,J)-13.5) = .5 THEN 400
500 Y(I,J) = Y(I,J)-10
510 FOR I1 = -1 TO 1
520 FOR J1 = -1 TO 1
530 Y(I+I1,J+J1) = Y(I+I1,J+J1)-1
540 NEXT J1
550 NEXT I1
560 NEXT J
570 PRINT
580 NEXT I
590 FOR I = L8+2 TO 22 :PRINT :NEXT I
600 PRINT TAB(20); "GENERATION"; G; TAB(39); "POPULATION"; Q; CHR$(07);
610 FOR I = L2-2 TO L8+2
620 FOR J = W2-2 TO W8+2
630 IF Y(I,J) = 0 THEN 660
640 Z(I,J) - Z(I,J)+Y(I,J)
650 Y(I,J) = 0
660 NEXT J
670 NEXT I
680 G = G+1
690 GOTO 280
```

RACET COMPUTES
702 Palmdale, Orange CA 92665

— RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET

FROM THE LEADER IN UTILITY SOFTWARE FOR THE TRS* COMPUTERS

★★ NEW ★★ HARD/SOFT DISK SYSTEM (MOD II) **$400**

The Hard Disk Software Implementation You Have Been Waiting For!! MOD II TRSDOS compatible — using Cameo controller interface to popular large hard disk fixed/removable combinations (Ampex, CDC, Diablo, Pertec, Wanco, etc.). Compatible with your existing programs — change only 'filename'. All disk BASIC statements identical. Improved dynamic file allocation. A single file can be as large as one disk — 20 megabytes or larger. Alternate mode allows 24-million byte record range. Directory expandable to handle thousands of files! Includes special XCOPY, DCS, and SZAP utilities for use with hard or soft disks. Parameterized FORMAT utility includes options for specifying the number of sectors/track, platters/drive, sectors/granule, sectors/directory, etc.

★★ NEW ★★ BASIC LINK FACILITY 'BLINK' **$25 Mod I**
(Mod I Min 32K 1-disk) **$50 Mod II**

Link from one BASIC program to another saving all variables! The new program can be smaller or larger than the original program in memory. The chained program may either replace the original program, or can be **merged** by statement number. The statement number where the chained program execution is to begin may be specified!

INFINITE BASIC (Mod I Tape or Disk) **$49.95**

Extends Level II BASIC with complete MATRIX functions and 50 more string functions. Includes RACET machine language sorts! Sort 1000 elements in 9 seconds!! Select only functions you want to optimize memory usage.

INFINITE BUSINESS (Requires Infinite BASIC) **$29.95**

Complete printer pagination controls — auto headers, footers, page numbers. Packed decimal arithmetic — 127 digit accuracy +, –, *, /. Binary search of sorted and unsorted arrays. Hash codes.

BASIC CROSS REFERENCE UTILITY (Mod II 64K) **$50.00**

SEEK and FIND functions for Variables, Line Numbers, Strings, Keywords. 'All' options available for line numbers and variables. Load from BASIC — Call with 'CTRL'R. Output to screen or printer!

DSM **$75.00 Mod I, $150.00 Mod II**
(Mod I Min 32K 2-drive system. Mod II 64K 1-drive)

Disk Sort/Merge for RANDOM files. All machine language stand-alone package for sorting speed. Establish sort specification in simple BASIC command File. Execute from DOS. Only operator action to sort is to change diskettes when requested! Handles multiple diskette files! Super fast sort times — improved disk I/O times make this the fastest Disk Sort/Merge available on Mod I or Mod II.

UTILITY PACKAGE (Mod II 64K) **$150.00**

Important enhancements to the Mod II. The file recovery capabilities alone will pay for the package in even one application! Fully documented in 124 page manual! XHIT, XGAT, XCOPY and superzap are used to reconstruct or recover data from bad diskettes! XCOPY provides multi-file copies, 'wild-card' mask select, absolute sector mode and other features. SUPERZAP allows examine/change any sector on diskette include track-0, and absolute disk backup/copy with I/O recovery. DCS builds consolidated directories from multiple diskettes into a single display or listing sorted by disk name or file name plus more. Change Disk ID with DISKID. XCREATE preallocates files and sets 'LOF' to end to speed disk accesses. DEBUGII adds single step, trace, subroutine calling, program looping, dynamic disassembly and more!!

DEVELOPMENT PACKAGE (Mod II 64K) **$125.00**

Includes RACET machine language SUPERZAP, Apparat Disassembler, and Model II interface to the Microsoft 'Editor Assembler Plus' software package including uploading services and patches for Disk I/O. Purchase price includes complete copy of Editor Assembler + and documentation for Mod I. Assemble directly into memory, MACRO facility, save all or portions of source to disk, dynamic debug facility (ZBUG), extended editor commands.

COMPROC (Mod I — Disk only) **$19.95**

Command Processor. Auto your disk to perform any sequence of instructions that you can give from the keyboard. DIR, FREE, pause, wait for user input, BASIC, NO OF FILES and MEM SIZE, RUN program, respond to input statements, BREAK, return to DOS, etc. Includes lowercase driver, debounce, screenprint!

Circle reader request for free 24-page catalog.

DEALER INQUIRIES INVITED

CHECK, VISA, M/C, C.O.D., PURCHASE ORDER
Telephone Orders Accepted (714) 637-5016

*TRS-80 is a registered trademark of Tandy Corporation

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes —

CIRCLE 188 ON READER SERVICE CARD

SORCERER SOFTWARE

SYSTEM 2 by Richard Swannell, loads into the top of available RAM and becomes an integral part of the BASIC language. All commands are single keystroke. SYSTEM 2 is written in Z80 and provides the following features:

1. **SCREEN EDITOR.** Use the editor to insert, replace, delete or rubout characters in your BASIC program. Watch the line change on the screen! Gone are the days of typing in a whole line to change one character!
2. **FUNCTION KEYS.** SYSTEM 2 allows 12 keys to be programmed to represent one or more characters or up to several lines of text each! After a key is programmed, by simply hitting key, all the text is sent to the processor just as if you typed it in on the keyboard! Function keys may be used in all modes of operation, including the editor. This feature is handy for lengthy and/or often used commands and may include multiple statements.
3. **RENUMBERING ROUTINE.** With a single keystroke your program is renumbered. Starting line number and increment may be changed.
4. **BASIC BUFFER PROTECTOR.** SYSTEM 2 sends a (CR) when the BASIC BUFFER is full. This prevents BASIC from crashing.
5. **PRINTER DRIVER.** Simply hit CTRL P to direct output to Centronics printer.
6. **RIVIVAL ROUTINE.** If NEW or CLOAD are typed, or RESET is hit by mistake, your program may be recovered. This is a safety device.

OTHER FEATURES

- RUNSTOP stops execution until any other key is hit.
- CLEAR clears screen then sends a (CR). Hit CLEAR to start on 'new page'
- CTRL characters such as ESC, LF and CLEAR don't return ?SN ERROR.
- RUB doesn't require the SHIFT key to be depressed. This quickens editing.
- Includes a Real Time Random Number Generator.
- Returns automatically to BASIC after TAPE CRC ERROR while CLOADing.
- Suppresses premature CRLF. Normally, if RUB is used extensively while typing in a BASIC line, the cursor will drop down to the next line before reaching the end of the current line. SYSTEM 2 prevents this.
- SYSTEM 2 requires 2K of memory and is available in 8, 16, 32 & 48K versions. **$35.50**

RS232 PRINTER DRIVER. Requires 250 bytes of memory and is relocatable. Suitable for MONITOR, STANDARD BASIC, WORD PROCESSOR PAC & DEVELOPMENT PAC. Stores each character in a buffer then sends the whole line at once, which solves timing problems. **$10.00**

COMBINED SYSTEM 2 & RS232 PRINTER DRIVER. SYSTEM 2 with the RS232 printer driver instead of Centronics printer driver. **$40.00**

LUNA LANDER. Written in Z80 and Basic and requiring 16K, LUNA LANDER uses graphics to the full. Land you craft on the moon in real time. But be careful to land softly, otherwise you will see your LANDER crumple before your very eyes! **$15.00**

SYSTEM SOFTWARE

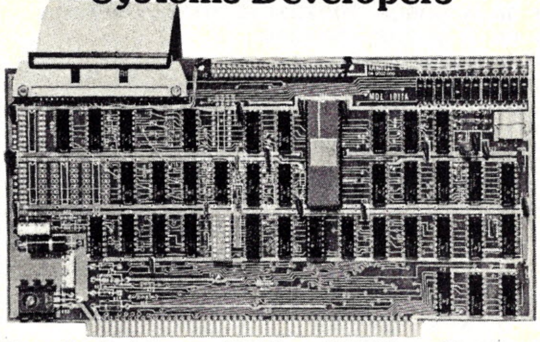
1 Kent Street, Bicton, 6157 Australia

Program comes on cassette and includes full documentation. Specify size of RAM. Prices in Australian Dollars. Add $2 for overseas airmail.
SORCERER is a trade mark of **EXIDY INC.**

CIRCLE 239 ON READER SERVICE CARD

Tarbell Floppy Disc Interface

Designed for Hobbyists and Systems Developers

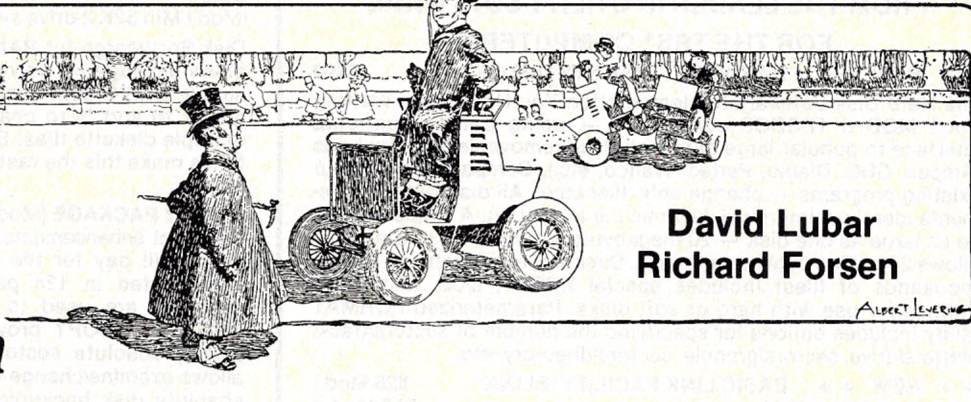


- Plugs directly into your IMSAI or ALTAIR and handles up to 4 standard single drives in daisy-chain.
- Operates at standard 250K bits per second on normal disc format capacity of 243K bytes.
- Works with modified CP/M Operating System and BASIC-E Compiler.
- Hardware includes 4 extra IC slots, built-in phantom bootstrap and on-board crystal clock. Uses WD 1771 LSI Chip.
- 6-month warranty and extensive documentation.

CIRCLE 202 ON READER SERVICE CARD

Tarbell Electronics
950 DOVLEN PLACE, SUITE B
CARSON, CA 90746
(213) 538-4251 • (213) 538-2254

CarRace: Anatomy of a Translation



David Lubar
Richard Forsen

CarRace came to us in the form shown in Listing #1, written for a DEC PDP/11. Since one of the most common areas of questions from readers concerns translation of programs, and since the program had to be translated anyway, this seemed like a good time to explain the techniques and approaches used when adapting a program to another language or dialect. The first, and most crucial, step is understanding exactly what is happening in the program. Attempting a blind line-by-line translation would be futile; you have to know what is happening and why it is happening, not just what is being performed on a line. Fortunately, Richard Forsen, who sent us this program, also sent in a number of notes on the functions used.

Richard Forsen, 9496 Weston Rd., New Hartford, NY 13413

So, what exactly does the program do? First, it sets up boundaries that mark the edges of the road and the position of the car. Next, the keyboard is checked for input. If the player does nothing, the program waits a certain amount of time, then makes a move. Random factors determine the way the road will curve, and the direction of the curve determines the type of character which will be used to mark the edge. Finally, a check is made to see if the driver has crashed. If he hasn't, his score is increased. This covers the main flow of the program. The next step is to find a way to achieve the same sequence of actions in another dialect of Basic.

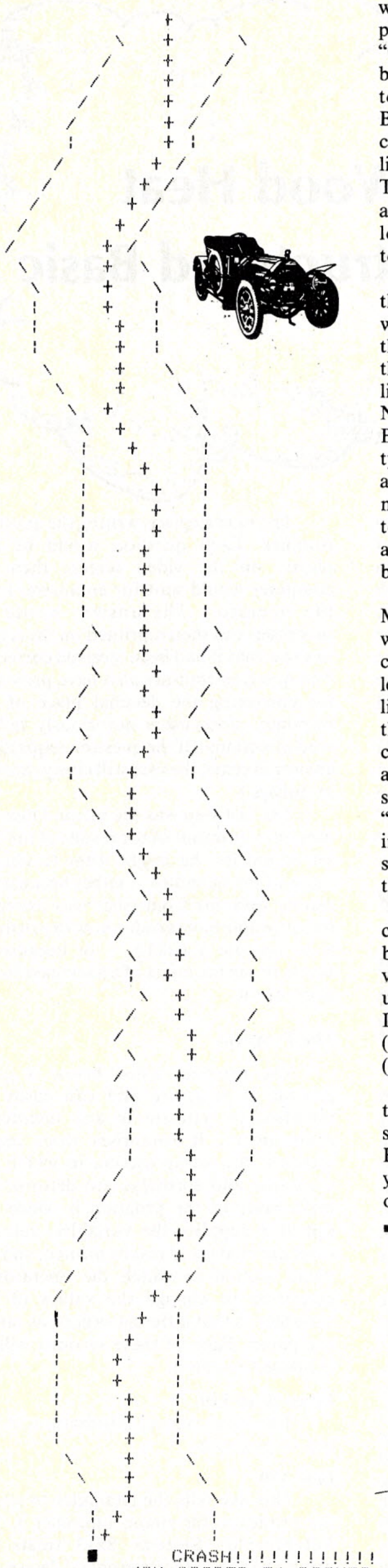
Listing #2 shows the translation, written, in this case, on a Commodore PET. Scanning the program, it becomes obvious that "B1" and "B2" control the width of the track. "T" is the position of the player, which is changed when he presses the "1" or "2" key. First, the DEC-specific functions have to be changed or removed. That means lines 1-30 have to go. Lines 40-60 are variable definitions, so they stay. Now we get to the tricky part. Lines 80-100

```
Ready

 1 RANDOMIZE
 10 OPEN "KB:" AS FILE 1
 20 ON ERROR GOTO 480
 30 A$=SYS(CHR$(3))
 40 B1=33
 50 B2=B1+10
 60 T=38
 80 B$=SYS(CHR$(4))
 90 GET #1
 100 WAIT 1
 110 FIELD #1,RECOUNT AS C$
 120 IF C$<>"1" AND C$<>"2" THEN 140
 130 ON VAL(C$) GOSUB 220,230
 140 IF B1=T OR B2=T THEN 400
 150 PRINT TAB(B1);D$;TAB(T);"+";TAB(B2);D$
 160 A=INT(RND*5)+1
 170 ON A GOSUB 200,210,210,210,200
 180 ON B GOSUB 250,240
 190 M=M+1:GOTO 80
 200 B=1:RETURN
 210 B=2:RETURN
 220 T=T-1:RETURN
 230 T=T+1:RETURN
 240 GOSUB 350:RETURN
 250 Y=INT(RND*3)+1
 260 IF X=Y THEN 250
 270 X=Y
 280 IF X=1 THEN D$="/"
 290 IF X=2 THEN D$="|"
 300 IF X=3 THEN D$="\"
 310 GOSUB 350
 320 RETURN
 350 B1=B1+X-2
 360 IF B1<1 THEN B1=1
 370 B2=B1+10
 380 IF B2>70 THEN B1=60:GOTO 370
 390 RETURN
 400 PRINT TAB(T);"■      CRASH!!!!!!!!!!!!!!!!"
 410 PRINT"YOU SCORED";M;"POINTS."
 460 GOTO 700
 480 RESUME 140
 700 END

Ready
```

Listing 1

```
40 B1=20
50 B2=B1+10
60 T=24
80 REM
90 GET C$
100 IFC$=""THEN S=S+1:IFS<2  THEN90
101 S=0
120 IFC$<>"1"ANDC$<>"2"THEN140
130 ON VAL(C$)GOSUB220,230
140 IF (B1=T) OR (B2=T) THEN 400
150 PRINT TAB(B1);D$;TAB(T);"+";TAB(B2);D$
160 A=INT(RND(1)*5)+1
170 ON A GOSUB 200,210,210,210,200
180 ON B GOSUB 250,240
190 M=M+1:GOTO80
200 B=1:RETURN
210 B=2:RETURN
220 T=T-1:RETURN
230 T=T+1:RETURN
240 GOSUB350:RETURN
250 Y=INT(RND(1)*3)+1
260 IFX=YTHEN250
270 X=Y
280 IF X=1THEND$="/"
290 IF X=2THEND$="I"
300 IFX=3THEND$="\"
310 GOSUB350
320 RETURN
350 B1=B1+X-2
360 IFB1<1THENB1=1
370 B2=B1+10
380 IFB2>39THENB1=29:GOTO370
390 RETURN
400 PRINT TAB(T);"※    CRASH!!!"
410 PRINT"YOU SCORED ";M;  "POINTS."
420 M=0
430 FORI=1TO500:NEXTI
460 GOTO 40
READY.
```

Listing 2



CRASH!!!!!!!!!!!!
YOU SCORED 76 POINTS.

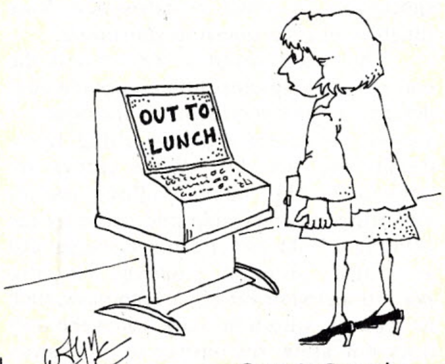
wait a specific time for an input. The input portion can be done in Microsoft with the "GET" statement, but the "WAIT" takes a bit more work. In this case, "GET" is made to loop back whenever there is a null input. By adding a counter to this loop, we can control the time. This is accomplished in line 100 (all changes refer to Listing #2). The limit on "S" needn't be "2"; it can be any value. The larger the number, the longer the wait. The "RND" functions had to be changed to fit the Microsoft format.

At this point, the game was working; the road appeared and scrolled upward while the cross moved within the lanes. But the game was written for a printer, where the width is 70. To use the game on a CRT, lines 40, 60 and 380 had to be scaled down. Now the road doesn't run off the screen. But the game ends fairly quickly, and typing "RUN" every few seconds can be annoying. To prevent this, the game was made into a loop. A delay allows the player to see his score, then the game begins again. "M," which holds the score, had to be reset to zero, as done in line 420.

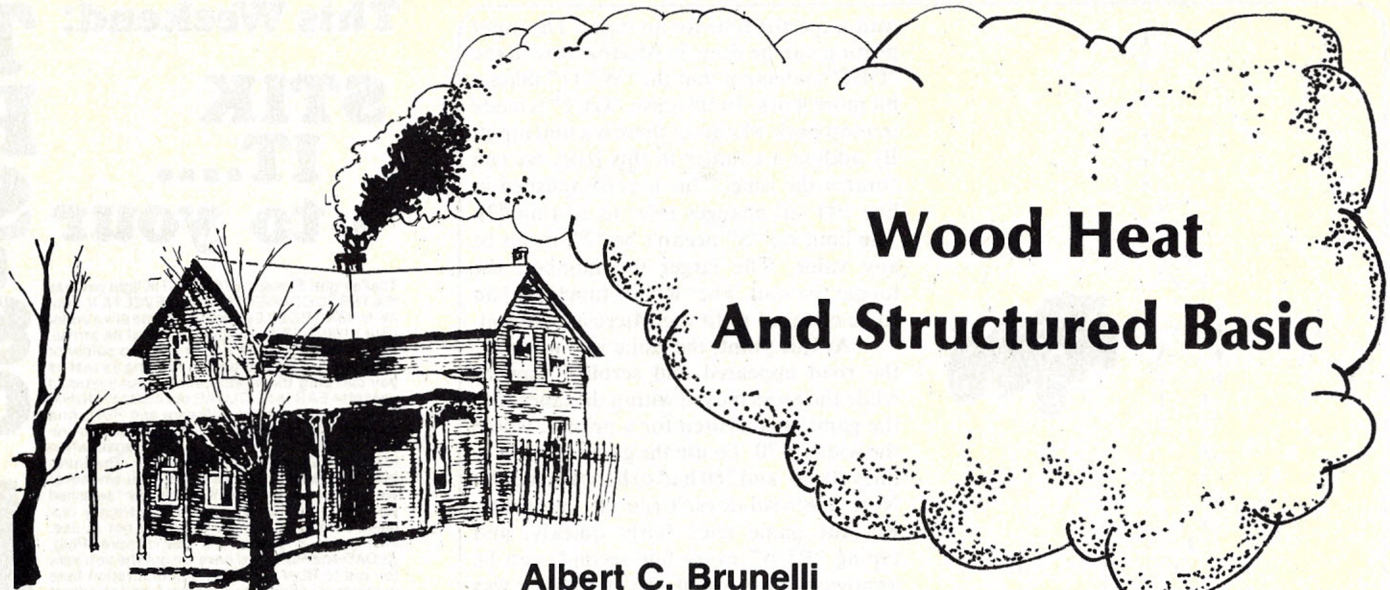
That about covers the translation. Many modifications are possible. First, wherever constants are found, variables can be substituted. This allows for skill levels and other changes. For example, in line 100, a variable could be used instead of the number "2." Then, with a routine to check the input, the player could use another key to "shift gears," changing his speed during the game. By changing the "10" in line 50, different skill levels can be introduced. The game can start with selection of difficulty; the harder the level, the narrower the road.

The major change in the program, combining "GET" with a delay loop, can be accomplished on other computers in various ways. TRS-80 owners will have to use the "INKEY$" function, Apple II Integer Basic users will need "C = PEEK (-16384)," followed by the usual "POKE (-16368,0)."

Once again, in doing any translation, the first task is to get a complete understanding of the logic behind the program. From there, it is merely a matter of getting your own language to do the same operations in the same order. □



© Creative Computing



Wood Heat And Structured Basic

Albert C. Brunelli

> *We publish lots of routines for checking user input, but wooden you think this is the ultimate?*

Introduction

The purpose of this article is twofold: to show how modular programming can be done in Basic, and to show how a small computer was used at an energy fair to help the visitors decide whether wood was an economically-viable alternative to fuel oil for home heating.

Background

When I was asked to write the program for an energy fair to be held at the New Hampshire Vocational Technical College in Berlin, I was given about one month's notice, and some very vague specifications as to what the program was expected to do. The woods department of the Brown Paper Company was to have a display of wood types with their associated BTU values per cord. My program was to complement that display and provide the visitors with a printout of wood values based on the cost of heating oil. That part was easy, as I had written a program for that purpose earlier.

The difficult part was the inclusion of the visitors' estimates on the costs associated with obtaining wood for fuel. These costs are variable, and depend strongly on the equipment used and its age and condition. I decided to set the program up so that the visitors could enter this information themselves. This decision made the program much more difficult to write.

I expected that most of the visitors would view not only the computer but even the keyboard as an exotic device. The program therefore had to be friendly. By friendly I mean that it must not display interpreter error messages and must prompt the operator in a way that he can understand. Since I expected that each visitor would want to compare the economy of heating with wood under different circumstances, I had to make provisions for changing the values of the variables so that several printouts would be available for comparison.

All these criteria taken together lead me to conclude that the program would have to be modular. The logic necessary to implement all the features mentioned would be fairly complex and the modular approach would make for easier design. Each question submitted to the operator would be in its own subroutine so that it could be called back for modification later. Each entry would be processed in a subroutine to avoid repeating the same function in each question subroutine.

The decimal-point problem was brought to my attention by a nine-year-old who was testing the program. He had two decimal points in one of his entries . . .

The first subroutine written was the entry processing subroutine, since it would determine the structure of all the others. I wanted to accept input in string form, despite the fact that all the entries are numerical. This method allows greater flexibility in processing and prevents the printing of interpreter error messages and associated program stoppage. It also permits a greater variety of responses, such as typing a question mark when the user does not know the answer to a question and wants the program to supply a default value.

The next step was writing the question routines. Each question would be displayed on the video screen; then the computer would wait for an answer from the operator. The answer would be processed and then reprinted on the screen to show that it had been received correctly. The question routines also have provision for correcting the decimal place of the response, since users notoriously fail to read directions. (If the question requests an answer in cents, the user will always answer in dollars.)

The last step was the menu. Once the user has a printout of the results of the first set of entries, he might want to change some of the variables — either because the results were far from what was expected (most people have no idea how expensive it is to operate a truck) — or because he wants to see the effect of an increase in the price of oil.

The Program

The program listed below is quite general in structure and can easily be modified to perform in any interactive environment. It comprises three major sections: the set-up section in which the constants and formulas are defined; the main body of the program is which the operator supplies the variables, calculations are made, and results printed; and the menu section in which the operator is permitted to change the values of the variables so that different situations can be compared. Each of these sections will be explained in detail.

Lines 10 to 110

The usual clerical information at the top of programs.

Line 120

This line tells the computer to attach the printer to file channel number 3. The PolyMorphic Systems 8813 treats the printer like any other peripheral device and talks to it through a file channel.

Albert C. Brunelli, Professor, Industrial Electronics, NHVTC, Berlin, Milan Road, Berlin, NH 03570.

Wood Heat, cont'd...

Lines 130 to 270

This section of the program sets up the constants we will use to determine the value of the wood. R is the ratio of the heat output per cord to the heat output per gallon of fuel oil. The burner efficiencies are included. The IE6 in line 160 is the million multiplier for the BTU values in lines 260 and 270. When the calculations are made, each BTU value will be multiplied by R to determine the gallon equivalent of each cord of wood. This gallon equivalent will then be multiplied by the cost of burning a gallon of oil to determine the value of a cord of wood.

The electrical cost factor for burning oil is P2. The 7.3 is the average current draw of the blower, igniter and pump. The 120 is the line voltage. The product of these two, which makes up the numerator of the fraction, is the power consumed while the burner is in operation. The denominator is the product of 1.5 and 1000. The 1.5 is the gallons per hour burning rate of the oil furnace and the 1000 is a conversion from watts to kilowatts, making the units of P2 kilowatt-hours/gallon. P2 will later be multiplied by the cost of electricity to determine the electrical cost of burning one gallon of oil.

The conditional in line 360 is there so that when the user assigns no value to his time, we do not bother with it.

Later, when all the results are printed, the cost per gallon will be multiplied by R and the BTU values of the woods to determine the gross dollar value of a cord of the given species. From this gross value will be subtracted the costs of getting the wood into the stove. These costs are entered by the operator.

Line 280

This line zeroes all the variables and is the re-entry point for multiple runs of the program.

Lines 290 to 370

Here we have the main body of the program. At this point we get all the information from the operator by going to each question subroutine.

The U appearing in lines 330 and 340 can have values 1, 2 or 3 depending on the method the operator chooses for obtaining fuelwood. Its value is set in the routine starting on line 1450. If the operator buys the wood delivered there is no additional hauling cost. If the wood is delivered cut and split there are no preparation costs.

The conditional in line 360 is there so that when the user assigns no value to his time, we do not bother with it, as it will not alter the value of the wood.

SYMBOL REFERENCE TABLE

Symbol Name	Purpose	Line #s
B	Array of Btu values of wood (millions of Btus per cord)	240,250,2800
C1	Electrical cost per gallon of fuel burned	2760, 2770
C2	Non-fuel costs of operating wood stove	2780, 2800
D9	Decimal point flag	1060, 1160, 1170
E1	Thermal efficiency of oil burner	140, 160, 2570
E2	Thermal efficiency of wood stove	150, 160, 2560
E	Error flag	530, 910, 1040, 1080, 1180, 1190, 1220, 1550, 1710, 1910, 2090, 2260, 2410
F1	Preparation cost per cord of wood	280, 620, 1730, 1780, 1790, 2620, 2630, 2780
H1	Transportation cost per cord	280, 620, 630, 1920, 1960, 1970, 2600, 2610, 2780
I	Dummy variable in FOR...NEXT loops	190, 250, 1090, 1100, 2520, 2740, 2750, 2790, 2800, 2860, 2870, 2890
O1	Oil price per gallon	280, 920, 980, 990, 2680, 2690, 2770
O2	Total fuel cost of oil per gallon	2770, 2800
O	Btu content of one gallon of fuel oil	130, 160
P1	Cost of electricity from utility per kilowatt-hour	280, 2420, 2480, 2490, 2580, 2590, 2760
P2	Electrical cost multiplier kilowatt-hours/gallon	170, 2760
Q	Question mark flag	540, 920, 1050, 1110, 1560, 1720, 1920, 2100, 2270, 2420
R	Gallons of oil to cords of wood conversion factor	160, 2800
S	Dummy variable used to carry entries back to calling routines in numerical form	550, 560, 580, 930, 960 970, 980, 1230, 1570, 1750, 1760, 1770, 1780, 1930, 1940, 1950, 1960, 2150, 2280, 2290, 2430, 2460, 2470, 2480
T1	Time spent working up the wood (hours per cord)	280, 2110, 2120, 2130, 2150, 2160, 2640, 2650, 2780
U	Flag for method of obtaining wood	280, 330, 340, 620, 630 650, 670, 1570, 1580, 1730, 2110, 2120, 2130
V	Dollar value of one cord of wood	2800, 2810, 2820, 2860, 2870
W1	Value of your time (dollars per hour)	280, 360, 2270, 2290, 2300, 2660, 2670, 2780
E$	Dummy string variable used to hold the individual characters entered by operatorin S$	1100, 1110, 1120, 1130, 1140, 1150, 1180, 1190, 1200
S1$	Holds acceptable characters in string from operator	1070, 1200, 1230
S$	String input from questions posed to operator	510, 890, 1070, 1100, 1220, 1530, 1690, 1890, 2070, 2240, 2390
W$	String array holding names of species of fuel wood	180, 190, 2860, 2870

Wood Heat, cont'd...

Lines 380 to 710

This section of the program is the menu. After the results of the first pass have been printed, control falls through to the menu. The choices are printed on the video screen and the user may choose whatever option is required. This feature of the program is essential if the user wishes to compare costs when market conditions change. An analysis of this section will show many of the principles of friendly programming.

Line 380 clears the video display (this is done at the beginning of each section to catch the attention of the operator). Lines 390 to 490 print the menu on the screen. Line 500 clears the keyboard buffer in case the user has been playing with the keys between operations.

Line 510 gets the user's choice in string form. The string is sent to the conversion routine which deletes unnecessary characters and checks for numerical entry.

Line 530 checks to see if the error flag has been set by the conversion routine (more about this later). If the error flag is set we go back to the menu for another try.

Line 540 checks for question mark entry (Q is the question mark flag). The question mark is not a reasonable response to the menu so we do an error message and go back to the menu display.

Lines 550 and 560 check to see that the response was within the permitted range. If it is not we tell the operator and go back for another try. If we are within the range we go off to the routines which modify the variables.

The subroutines which request the entry of variables work similarly. The question is printed on the screen, the answer is accepted as a string, and the string is converted to a number or a flag is set. If the Q flag is set we supply a default value for the variable. If the error flag is set, an error message is printed and then a new response is requested. When there is no flag set we check for the proper decimal place in the number, correct it if necessary, and then print the result on the screen. Line 990 shows how the number is printed on the screen after processing. The %$ stuff is a format statement which will cause the variable to be printed in a field right justified, eight characters wide with three places after the decimal point and a dollar sign to the left of the number.

String Conversion Routine, Lines 1040 to 1240

Lines 1070 and 1080 check to see if just a carriage return was pressed. If so, the error flag is set and control is passed to the error message routine.

The FOR . . . NEXT loop does most of the work in this routine. Each character in the string is checked for validity. A question mark sets the Q flag and returns control immediately. The EXIT in line 1110 clears the FOR . . . NEXT stuff from the stack when the loop is terminated early. Dollar signs, minus signs, blanks and all but the first decimal place are discarded.

The decimal-point problem was brought to my attention by a nine-year-old who was testing the program. He had two decimal points in one of his entries and got an interpreter error message on line 1230 because the string could not be converted with two decimal points in it.

Lines 1180 and 1190 set the error flag if the character is not numerical. Once again we use the EXIT to clear the stack. Line 1200 rebuilds the string with acceptable characters.

If the rebuilt string is null, line 1220 detects the condition and sets the error flag. If the entry has passed all the tests, it is converted to a numerical variable and returned to the calling routine. We must test for a null string in line 1220 because it is possible that someone might fill the string with spaces. If the string is full of decimal points we still have a problem, and if I were to write the program over, I might strip the decimal points in this routine and correct for it in the calling routine. There were no problems with this during its use at the energy fair.

I expected that most of the visitors would view not only the computer but even the keyboard as an exotic device. The program therefore had to be friendly.

The printout routine is located beginning at line 2520. If you recall we opened file channel 3 for the printer back in line 120 and now whenever we PRINT: 3 we send data through this channel to the printer. If we wish to display data on the screen we just use the PRINT without the colon. Lines 2550 through 2710 print out the value of the variables used in the calculations so that each printout of wood values will have with it the values of the variables used in its calculation. When referring to the printouts at a later time, we will know what information was used to obtain them.

Lines 2760 through 2810 perform the calculations of wood value. Line 2760 calculates the cost of electricity used to burn one gallon of fuel oil. P1 is the cost of one kilowatt-hour of electricity and P2 is the number of kilowatt-hours/gallon of electrical energy consumer per gallon of oil burned.

Line 2770 calculates the true cost of one gallon of fuel oil by adding the purchase price of the oil to the electrical cost of burning it.

Line 2780 calculates the cost of burning one cord of firewood. C2 is the sum of the hauling cost (H1), the preparation cost (F1), and the value of the labor involved (T1*W1).

Line 2800 calculates the value of a cord of each species of interest. The value is determined by multiplying the cost per gallon of oil (02) times the BTU-efficiency ratio (R) times the BTU content of one cord of the species of interest. From this product is subtracted the cost of getting the wood into the stove (C2). The result is the actual dollar value of a cord of that species given all the costs entered by the operator. It is the price you can afford to pay for firewood and come out even. If you pay less than the calculated value for wood you are better off heating with wood. If you pay more you should heat with oil.

Conclusion

I hope I have shown here that Basic can be used to create structured, modular programs that are in some ways superior to those created by the more naturally structured languages. Admittedly, programs in those grander languages, Pascal and C, can be easy to decipher if the function names are well chosen to reflect the nature of the function being called. But if the reader of the program wants to know exactly how the function works he will have a difficult time finding it in a long listing. The use of remarks in Basic enables the reader to determine the purpose of the subroutine, and the line number associated with the GOSUB makes it easy to locate the routine when more detailed information is required. □

```
10 REM
20 REM PROGRAM TO DETERMINE VALUE OF WOOD AS HEATING FUEL
30 REM
40 REM          A. BRUNELLI
50 REM          NHVTC, BERLIN
60 REM          MILAN RD.
70 REM          BERLIN, NH 03570
80 REM
90 REM       WRITTEN IN POLYMORPHICS BASIC VER. B08
100 REM      10/20/79
110 REM
120 FILE:3,LIST
130 O=140000\REM BTU CONTENT OF ONE GAL. #2 FUEL OIL
140 E1=0.65\REM EFFICIENCY OF OIL BURNER
150 E2=0.50\REM EFFICIENCY OF AIR TIGHT WOOD STOVE
160 R=E2*1E6/(E1*O)\REM CONVERSION FACTOR GAL. TO CORD
```

Wood Heat, cont'd...

```
170 P2=7.3*120/1500\REM ELECTRICAL COST MULTIPLIER
180 DIM W$(15:15)\REM * HOLD SPECIES NAMES
190 FOR I=1 TO 15\READ W$(I)\NEXT
200 DATA"Hophornbeam","Beech","Sugar maple","Red oak","Yellow Birch"
210 DATA"White ash","Red Maple","White birch","Elm","Red Spruce"
220 DATA "Pin cherry","Poplar","Balsam fir","White pine"
230 DATA "Basswood"
240 DIM B(15)\REM BTU/CORD OF SPECIES ABOVE IN MILLIONS
250 FOR I=1 TO 15\READ B(I)\NEXT
260 DATA 24.7,21.8,21.8,21.7,21.3,20,19.1,18.2
270 DATA 17.7,15,14.2,14.1,13.5,13.3,12.6
280 U,T1,W1,P1,O1,F1,H1=0
290 GOSUB 720\REM * EXPLAIN PROGRAM AND METHOD OF DATA ENTRY
300 GOSUB 850\REM * GET CURRENT FUEL COST
310 GOSUB 2350\REM * GET COST OF ELECTRICITY
320 GOSUB 1450\REM * GET METHOD OF OBTAINING WOOD
330 IF U=3 THEN GOSUB 1840\REM * GET HAULING COST
340 IF U>1 THEN GOSUB 1630\REM * GET PREP. COST
350 GOSUB 2200\REM * GET VALUE OF TIME
360 IF W1=0 THEN GOSUB 2110 ELSE GOSUB 2020\REM * TIME SPENT
370 GOSUB 2520\REM * PRINT RESULTS
380 PRINT CHR$(12)
390PRINT TAB(10),"What would you like to change ?"
400PRINT"#2 fuel oil cost",TAB(35),"1"
410PRINT"Cost of electricity",TAB(35),"2"
420PRINT"Method of obtaining wood",TAB(35),"3"
430PRINT"Hauling cost",TAB(35),"4"
440PRINT"Preparation cost",TAB(35),"5"
450PRINT"Time taken",TAB(35),"6"
460PRINT"Value of time",TAB(35),"7"
470PRINT"Printout",TAB(35),"8"
480PRINT"Run program again",TAB(35),"9"
490PRINT"Terminate program",TAB(35),"10"
500 PRINT CHR$(24)\REM * CLEAR KEYBOARD BUFFER
510PRINT\INPUT"Number of choice ?      ",S$
520 GOSUB 1040\REM * CONVERT STRING
530 IF E=1 THEN 500
540 IF Q=1 THEN 570
550 S=INT(S)
560 IF S>0 THEN IF S<11 THEN 580
570 GOSUB 1250\GOSUB 1380\GOTO 380
580 ON S GOTO 590,600,610,650,670,690,700,710,280,2920
590 GOSUB 850\GOTO 380\REM * NEW FUEL PRICE
600 GOSUB 2350\GOTO 380\REM * NEW ELECT. PRICE
610 GOSUB 1450\REM * NEW METHOD OF GETTING WOOD. REQUIRES OTHER CHANGES
620 IF U=1 THEN H1=0\F1=0\GOSUB 2020\GOTO 380
630 IF U=2 THEN H1=0\GOSUB 1630\GOSUB 2020\GOTO 380
640 GOSUB 1840\GOSUB 1630\GOSUB 2020\GOTO 380
650 IF U=3 THEN 660
660 GOSUB 1840\GOTO 380
670 IF U<>1 THEN GOSUB 1630\GOTO 380
680PRINT"* YOU DON'T HAVE ANY PREP COST *"\GOSUB 1370\GOTO 380
690 GOSUB 2020\GOTO 380
700 GOSUB 2200\GOTO 380
710 GOSUB 2520\GOTO 380
720 PRINT CHR$(12)
730PRINT TAB(8),"This program will help you determine the economy"
740PRINT"of heating with wood. I will ask you some questions about"
750PRINT"your energy costs and use your answers to determine the dollar"
760PRINT"values of cords of various wood species."
770PRINTTAB(5),"Please enter your answers by typing them on the keyboard."
780PRINT"If you make an error press the key marked 'DELETE' on"
790PRINT"the right side of the keyboard. If you do not know the"
800PRINT"answer to a question enter a question mark (?)."
810PRINT"To conclude your entry press the key marked 'RETURN'"
820PRINT"on the right side of the keyboard."
830 GOSUB 1370\REM * WAIT FOR KEYPRESS
840 RETURN
850 PRINT CHR$(12)
860PRINT"         Please enter the price you currently pay for"
870PRINT"#2 fuel oil delivered to your home."
880 PRINT CHR$(24)\REM * CLEAR KEYBOARD BUFFER
890 PRINT\INPUT" Current oil price (cents/gal.) ?  ",S$
900 GOSUB 1040\REM * CONVERT STRING
910 IF E=1 THEN 880
920 IF Q=1 THEN O1=0.89\GOTO 990
930 IF S<>0 THEN 960
940PRINT"If oil were free we wouldn't be here today...."
950 GOTO 880
960 IF S<10 THEN S=S*100\GOTO 960\REM * CORRECT FOR DOLLAR ENTRY
970 IF S>1000 THEN S=S/10\GOTO 970\REM THERE ARE LIMITS
980 O1=S/100\REM * CONVERT TO DOLLARS
990PRINT"Fuel cost",TAB(30),%$8F3,O1,"/gal."
1000 GOSUB 1370\REM * WAIT FOR KEYPRESS
1010 RETURN
1020 REM
1030 REM * CHAR TO NUM CONVERSION ROUTINE
1040 E=0\REM * ERROR FLAG FOR NON NUMERICAL CHAR
1050 Q=0\REM * QUESTION MARK FLAG
1060 D9=0\REM * FLAG FOR # OF DECIMAL POINTS
1070 L=LEN(S$)\S1$=""\REM * CLEAR S1$
1080 IF L=0 THEN E=1\RETURN
```

```
1090 FOR I=1 TO L
1100 E$=MID$(S$,I,I)
1110 IF E$="?" THEN Q=1\EXIT 1240
1120 IF E$="$" THEN 1210\REM * IGNORE $,-,SPACE
1130 IF E$="-" THEN 1210
1140 IF E$=" " THEN 1210
1150 IF E$<>"." THEN 1180
1160 IF D9=1 THEN 1210\REM * IGNORE EXTRA DEC. POINTS
1170 D9=1\GOTO 1200\REM * INCLUDE ONLY ONE DEC. POINT
1180 IF E$<"0" THEN E=1\EXIT 1250
1190 IF E$>"9" THEN E=1\EXIT 1250
1200 S1$=S1$+E$
1210 NEXT
1220 IF LEN(S$)=0 THEN E=1\GOTO 1250
1230 S=VAL(S1$)
1240 RETURN
1250 ON RND(5) GOTO 1260,1280,1300,1320,1340
1260PRINT" * I'm not that stupid. ",
1270 GOTO 1350
1280PRINT" * DOES NOT COMPUTE. ",
1290 GOTO 1350
1300PRINT" * HUH ??..... ",
1310GOTO 1350
1320PRINT"Je ne comprends pas.......",
1330 GOTO 1350
1340PRINT"I sincerely doubt that is what you intended. ",
1350 PRINT" TRY AGAIN."
1360 RETURN
1370PRINT
1380 PRINT TAB(8),"Press 'RETURN' to continue..."
1390 PRINT CHR$(24)\REM * CLEAR INPUT BUFFER
1400 IF INP(0)=0 THEN 1400
1410 PRINT CHR$(24)\ REM * CLEAR INPUT BUFFER
1420 RETURN
1430 REM *
1440 REM * ROUTINE TO DETERMINE METHOD OF OBTAINING FUEL WOOD
1450 PRINT CHR$(12)
1460PRINT"      Choose the method below which is closest to the"
1470PRINT"method you use to obtain firewood."
1480 PRINT
1490 PRINT"Delivered cut, split and stacked.......  1"
1500 PRINT"Delivered unsplit......................  2"
1510 PRINT"Fell trees and haul logs...............  3"
1520 PRINT CHR$(24)\REM * CLEAR KEYBOARD BUFFER
1530 PRINT\INPUT"Your choice ?.........   ",S$
1540 GOSUB 1040
1550 IF E=1 THEN 1520
1560 IF Q=1 THEN 1590
1570 U=INT(S+.5)
1580 IF U>0 THEN IF U<4 THEN 1600
1590 GOSUB 1250\GOTO 1520
1600 RETURN
1610 REM
1620 REM * PREPARATION COSTS PER CORD
1630 PRINT CHR$(12)
1640PRINT"      Please enter the cost of working up one cord of firewood"
1650PRINT"into stove sized pieces. Please include the costs of gas,"
1660PRINT"oil, files, chains, saw, mauls, wedges, rental of power"
1670PRINT"splitter and other out of pocket costs for preparing the wood."
1680 PRINT CHR$(24)\REM * CLEAR KEYBOARD BUFFER
1690PRINT\INPUT" Preparation costs (dollars/cord) ?  ",S$
1700 GOSUB 1040
1710 IF E=1 THEN 1680
1720 IF Q=0 THEN 1750
1730 IF U=2 THEN F1=2.25 ELSE F1=3
1740 GOTO 1790
1750 IF S=0 THEN GOSUB 1250\GOTO 1680
1760 IF S<1 THEN S=S*10\GOTO 1760 \ REM * TOO CHEAP
1770 IF S>100 THEN S=S/10\GOTO 1770\REM * TOO EXPENSIVE
1780 F1=S
1790PRINT\PRINT"Preparation costs",TAB(20),%$8F2,F1
1800 GOSUB 1370
1810 RETURN
1820 REM
1830 REM * ROUTINE TO DETERMINE HAULING COST
1840 PRINT CHR$(12)
1850PRINT"      Please enter the out of pocket costs of transporting"
1860PRINT"one cord of wood to its final location. Include the costs"
1870PRINT"of gas, oil, and truck depreciation or rental."
1880 PRINT CHR$(24)\REM * CLEAR KEYBOARD BUFFER
1890PRINT\INPUT"Transportation cost (dollars/cord) ?  ",S$
1900 GOSUB 1040
1910 IF E=1 THEN 1880
1920 IF Q=1 THEN H1=15\GOTO 1970
1930 IF S=0 THEN 1960
1940 IF S<1 THEN S=S*10\GOTO 1940
1950 IF S>100 THEN S=S/10\GOTO 1950
1960 H1=S
1970PRINT\PRINT"Transportation cost",TAB(20),%$8F2,H1
1980 GOSUB 1370
1990 RETURN
2000 REM
2010 REM * SUBROUTINE TO DETERMINE TIME SPENT
2020 PRINT CHR$(12)
```

Wood Heat, cont'd...

```
2030PRINT"      Please enter the number of hours you spend handling"
2040PRINT"each cord of wood. Be sure to include time spent getting"
2050PRINT"the wood into the stove and cleaning the stove and flue."
2060 PRINT CHR$(24)\REM * CLEAR KEYBOARD BUFFER
2070 PRINT\INPUT"Time spent (hours/cord) ?  ",S$
2080 GOSUB 1040
2090 IF E=1 THEN 2060
2100 IF Q=0 THEN 2150
2110 IF U=1 THEN T1=2\REM * BUYS CUT AND SPLIT
2120 IF U=2 THEN T1=5\REM * BUYS IN 4' OR 8' LENGTHS
2130 IF U=3 THEN T1=8\REM * CUTS THE TREES
2140 GOTO 2160
2150 T1=S
2160PRINT\PRINT"Hours per cord",TAB(20),%7F2,T1
2170 GOSUB 1370
2180 RETURN
2190REM * ROUTINE TO DETERMINE VALUE OF TIME
2200 PRINT CHR$(12)
2210 PRINT"      Please enter what you consider to be the value"
2220PRINT"of your time spent dealing with firewood."
2230 PRINT CHR$(24)\REM * CLEAR KEYBOARD BUFFER
2240 PRINT\INPUT"Value of time (dollars/hour) ?     ",S$
2250 GOSUB 1040
2260 IF E=1 THEN 2230
2270 IF Q=1 THEN W1=4\GOTO 2300\REM * DEFAULT VALUE OF TIME
2280 IF S>100 THEN S=S/10\GOTO 2280
2290 W1=S
2300PRINT\PRINT"Value of time",TAB(20),%$6F2,W1
2310 GOSUB 1370
2320 RETURN
2330 REM
2340 REM* ROUTINE TO DETERMINE ELECTRICITY COST
2350 PRINT CHR$(12)
2360PRINT"      Please type in the price you now pay for"
2370PRINT"electricity."
2380 PRINT CHR$(24)\REM * CLEAR KEYBOARD BUFFER
2390 PRINT\INPUT"Cost of electricity (cents/KWhr) ?   ",S$
2400 GOSUB 1040
2410 IF E=1 THEN 2380
2420 IF Q=1 THEN P1 = .069\GOTO 2490\REM * DEFAULT COST (N.H.)
2430 IF S<>0 THEN 2460
2440 PRINT"If electricity were free you wouldn't heat with wood."
2450 GOTO 2380
2460 IF S<1 THEN S=S*10\GOTO 2460
2470 IF S>100 THEN S=S/10\GOTO 2470
2480 P1=S/100
2490PRINT\PRINT"Electricity cost",TAB(20),%$8F3,P1,"/KWhr"
2500 GOSUB 1370
2510 RETURN
2520 FOR I=1 TO 4\PRINT:3,CHR$(10)\NEXT
2530PRINT CHR$(12)
2540 PRINT:3,TAB(10),"ENERGY FAIR  Oct. 20,1979"
2550 PRINT:3,CHR$(10)
2560PRINT:3,"Efficiency of wood stove",TAB(35),E2*100," %"
2570PRINT:3,"Efficiency of oil burner",TAB(35),E1*100," %"
2580PRINT:3,"Cost of electricity",TAB(30),%$8F3,P1,"/KWhr"
2590PRINT"Cost of electricity",TAB(30),%$8F3,P1,"/KWhr"
2600PRINT:3,"Transportation cost",TAB(30),%$8F2,H1,"/cord"
2610PRINT"Transportation cost",TAB(30),%$8F2,H1,"/cord"
2620PRINT:3,"Preparation cost",TAB(30),%$8F2,F1,"/cord"
2630PRINT"Preparation cost",TAB(30),%$8F2,F1,"/cord"
2640PRINT:3,"Time spent",TAB(30),%8F2,T1,"Hrs/cord"
2650PRINT"Time spent",TAB(30),%8F2,T1,"Hrs/cord"
2660PRINT:3,"Value of time",TAB(30),%$8F2,W1,"/hour"
2670PRINT"Value of time",TAB(30),%$8F2,W1,"/hour"
2680PRINT:3,"Cost of fuel oil",TAB(30),%$8F3,O1,"/gal."
2690PRINT"Cost of fuel oil",TAB(30),%$8F3,O1,"/gal."
2700PRINT:3,CHR$(10)
2710 PRINT
2720 PRINT:3,"Wood type",TAB(30),"Value of one cord (20 % Moisture)"
2730PRINT"Wood type",TAB(30),"Value of one cord"
2740FOR I=1 TO 48\PRINT:3,"-",\NEXT\PRINT:3,CHR$(13)
2750FOR I=1 TO 48\PRINT "-",\NEXT\PRINT
2760 C1=P1*P2\REM * COST OF ELECTRICITY TO RUN OIL BURNER
2770 O2=O1+C1
2780 C2=H1+F1+T1*W1\REM * COSTS OF OPERATION FOR WOOD HEAT
2790 FOR I=1 TO 15
2800 V=O2*R*B(I)-C2\REM * VALUE OF ONE CORD OF WOOD
2810 V=INT(V+.5)
2820 IF V>-100 THEN 2860
2830 PRINT:3,"You would lose quite a bundle....."
2840PRINT"You would lose quite a bundle....."
2850 EXIT 2890
2860 PRINT:3,W$(I),TAB(33),%$8I,V
2870PRINT W$(I),TAB(33),%$8I,V
2880 NEXT
2890 FOR I=1 TO 4\PRINT:3,CHR$(10)\NEXT
2900GOSUB 1370
2910 RETURN
2920 PRINT CHR$(12)
2930 PRINT\PRINT\PRINT
2940PRINT"Have a mild winter............"
```

ENERGY FAIR Oct. 20,1979

Efficiency of wood stove	50 %
Efficiency of oil burner	65 %
Cost of electricity	$.070/KWhr
Transportation cost	$.00/cord
Preparation cost	$.00/cord
Time spent	3.00Hrs/cord
Value of time	$3.50/hour
Cost of fuel oil	$.960/gal.

Wood type	Value of one cord (20 % Moisture)
Hophornbeam	$125
Beech	$109
Sugar maple	$109
Red oak	$109
Yellow Birch	$107
White ash	$99
Red Maple	$95
White birch	$90
Elm	$87
Red Spruce	$72
Pin cherry	$68
Poplar	$67
Balsam fir	$64
White pine	$63
Basswood	$59

ENERGY FAIR Oct. 20,1979

Efficiency of wood stove	50 %
Efficiency of oil burner	65 %
Cost of electricity	$.070/KWhr
Transportation cost	$.00/cord
Preparation cost	$2.50/cord
Time spent	8.00Hrs/cord
Value of time	$3.50/hour
Cost of fuel oil	$.960/gal.

Wood type	Value of one cord (20 % Moisture)
Hophornbeam	$105
Beech	$89
Sugar maple	$89
Red oak	$89
Yellow Birch	$87
White ash	$79
Red Maple	$75
White birch	$70
Elm	$67
Red Spruce	$52
Pin cherry	$48
Poplar	$47
Balsam fir	$44
White pine	$43
Basswood	$39

ENERGY FAIR Oct. 20,1979

Efficiency of wood stove	50 %
Efficiency of oil burner	65 %
Cost of electricity	$.070/KWhr
Transportation cost	$18.00/cord
Preparation cost	$3.00/cord
Time spent	12.00Hrs/cord
Value of time	$3.50/hour
Cost of fuel oil	$.960/gal.

Wood type	Value of one cord (20 % Moisture)
Hophornbeam	$73
Beech	$57
Sugar maple	$57
Red oak	$56
Yellow Birch	$54
White ash	$47
Red Maple	$42
White birch	$37
Elm	$34
Red Spruce	$19
Pin cherry	$15
Poplar	$15
Balsam fir	$11
White pine	$10
Basswood	$6

H&E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

★ *All Orders processed within 24-Hours*

★ *30-Day Money Back Guarantee on all Software (less a $3 penalty for handling)*

★ *10-Day Money Back Guarantee on Disk Drives and Printers* **PLUS** *120-Days Free Service*

- **LEARNING LEVEL II** By David Lien
  The Original Author Of The Level Manual
  A Step By Step approach to Learning Level II especially geared to new TRS-80™ Owners **$15.95**

- **TRS-80™ DISK AND OTHER MYSTERIES**
  Over 100 pages of indespensible information for disk owners. Learn to recover information from bad disks, how to make Basic programs unlistable and 12 more chapters of never published tips and information. Written by H.C. Pennington.
  (For all Disk Owners). **$22.50**

NEW SBSG BUSINESS SYSTEM FOR MODEL I OR MODEL II - IN STOCK
- General Ledger
- Accounts Receivable
- Accounts Payable
- Payroll
- Inventory Control with Invoicing

- Each module can be operated individually or as a coordinated **SYSTEM.** Turn-Key error catching operation for beginners.
- Complete manual and documentation accompany each program.
- Minimum System requirements 2-Disk Drives for Model I...1-Disk Drive for Model II
- Each module can be formatted to span data on up to 4-Disk Drives
- Free 30-Day telephone consultation
- Call for complete specifications
- Model I Version **$125.00** Per Module, **$495.00** Per System
- Model II Version **$225.00** Per Module, **$995.00** Per System

DATA MANAGEMENT SYSTEMS
- DMS replace index cards or any data requiring long lists of information.

- TBS In-Memory Information System (For Cassette Systems) **$39.95**
- TBS Disk Data Manager (Requires 1 or more disk drives)...Set up fast random access, files in minutes. Stores up to 320K of information on 4 Drives. Up to 10 fields and 255 characters per record. Supports upper and lower case. RS-232 or TRS-232...Features complete editing **$49.50**
- Personal Software CCA Data Management System...Completely user oriented, menu drive, 130 page Step By Step Manual...Capable of inventory control, sorting data, reporting data in nearly any form (for reports and mailing labels). Sorts data by up to 10 fields for zip code, balance due, geographic location or whatever. Prints reports with subtotals and totals automatically calculated. Fast random access **$75.00**

FROM RACET COMPUTES
- **REMODEL-PROLOAD** - Renumbers program lines, combines programs. The only renumber program that will renumber the middle of a program. Specify 16K, 32K or 48K. Works with Cassette or Disk **$34.95**
- **GSF** - Use in your Basic Programs for Instant Sorting (will sort 1000 items in 9 seconds). Other commands include Compress and Uncompress Data. Duplicate Memory, Display Screen Controls and Fast Graphic Controls **$24.95**
  (For Cassette or Disk, specify 16K, 32K or 48K).
- **DOSORT** - All G.S.F. commands plus special Multiple Disk Sorting Routines **$34.95**
  (Specify 32K or 48K)
- **INFINITE BASIC** - Adds 70 commands to your TRS-80™ including Instant Sort, Matrix Commands, String Commands, Left and Right Justification, String Centering, Simultaneous Equations, Upper and Lower Case Reverse and more. (For Cassette or Disk) **$49.95**
- **INFINITE BUSINESS** (Requires Infinite Basic) Eliminate Round-off error, 127-Digit Calculation Accuracy, Insert New Elements in Sorted Arrays, Automatic Page Headings, Footings and Pagination, Multiple Precision Arithmetic and more. (For Cassette or Disk) **$29.95**
- **COPSYS** - Copy Machine Language Programs (For Cassette Only) **$14.95**
- **DSM** (Disk Sort Merge) **$75.00**

FROM SMALL SYSTEM SOFTWARE
- **RSM-2** Machine Language Monitor **$26.95**
- **RSM-2D** Disk Version of RSM-2 **$29.95**
- **DCV-1** Converts Machine Language Programs from tape to disk **$9.95**
- **AIR RAID** - The ultimate TRS-80™ game converts your TRS-80™ into a real time shooting gallery **$14.95**
- **BARRICADE** - A fast pong style game **$14.95**
- **CPM** - (For Disk Only) **$150.00**
- **TRS-232 INTERFACE** - Interface with Software driver RS-232 printers to your TRS-80™ **$49.95**
- **TRS-232 FORMATTER** - Additional (optional) Software for TRS-232 owners. Adds many printer commands to your TRS-80™ **$14.95**
  (With purchase of TRS-232) **$9.95**
- **PENMOD** - Use the Electric Pencil with RS's lower case modification **$19.95**

FROM GALACTIC SOFTWARE
- **MAIL PAC** - For Model I Disk Systems only **$99.95**
  Quick-sorting full user control over mailing list from Galactic Software.
- **STOCK MARKET PAC** **$99.95**

FROM APPARAT NEW DOS + **$99.95**
35, 40 and 77 Track Versions available.
- **NEW DOS/80** (With variable record length files, chainings and many other features **$149.95**

FROM THE BOTTOM SHELF
- **CHECKBOOK II** (For Cassette or Disk) **$39.95**
- **SYSTEM DOCTOR** (A complete diagnosis of your TRS-80™...Checks memory, video, cassette, disk, ROM, and all other parts of your system) For Cassette or Disk **$28.50**
- **CHECKBOOK REGISTER ACCOUNTING SYSTEM** (Requires 2 disk drives) **$75.00**
- **LIBRARY 100** - 100 established business, game and educational programs plus **FREE** Tiny Pilot all for **$49.50**
- **BASIC TOOL KIT** - Lists all variables, GOTO's and GOSUB's in your program **$19.80**

- **SOUNDWARE** - Adds sound to your TRS-80™ Just plus it in **$29.95**
  Sample programs included.
- **TING TONG** - Can be used with Soundware for a **Sound** version of pong **$9.95**

- **VIC - The Carta Visual Instructional Computer Program** **$19.95**
  The Level II 16K Cassette is designed to teach beginners the Basics of Machine Language and Assembly Language Programming. See every Machine Language Instruction Display on your video. VIC includes Step By Step 55 page manual

VISTA V80 DISK DRIVE - 110K of Storage **$395.00**
Add $29.95 for Cable (Free with purchase of 2-Disk Drives). 10 day money back guarantee.

FROM HOWE SOFTWARE
- **MON-3** - Machine Language Programming for beginners. MON-3 is a complete System Monitor with Users Manual **$39.95**
- **MON-4** - Disk Version of MON-3 **$49.95**

FROM MICROSOFT
- **LEVEL III BASIC** **$49.95**
  Now Cassette owners can add Disk Commands to their TRS-80™ without owning a Disk Drive
- **MICROSOFT DISK ADVENTURE** **$29.95**
- **TRSDOS BASIC COMPILER** **$195.00**
  Run Basic Programs up to 15 times faster.

- **NEC BUSINESS QUALITY PRINTERS** (For MOD-I or MOD-II) **$2,995.00**

THE ELECTRIC PENCIL

Cassette	$99.95
Disk	$150.00
MOD-II Version	$325.00

- **HORSE SELECTOR II** By Dr. Hal Davis
  The TRS-80™ version updated for the TRS-80™ and originally reviewed in Systems and Methods **$50.00**

••• EVERYTHING ••• FOR MOD-II OWNERS

NEW MOD-II NEWSLETTER MOD-II Catalog **Free** w/subscription	**$12/year**
MAIL PAC	**$199.95**
MICROSOFT BASIC COMPILER	**$395.00**
MICROSOFT BASIC	**$325.00**
GSF SORT ROUTINE	**$50.00**
CP/M	**$170.00**
PEACHTREE BUSINESS SOFTWARE	**Call**
WORD STAR	**$495.00**

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

HOURS: 9-5
Monday thru Saturday

master charge | **24 HOUR ORDER LINE** | VISA

(914) 425-1535

48-Page Catalog $2 FREE With Any Order
Order By Phone Or Mail
Add $1 Per Order For Shipping Within UPS Areas
Add $3 For C.O.D.
Add $3 For All Foreign And Non-UPS Shipments
Add $3 For UPS Blue Label

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

Planning and Evaluating Your Diet

Douglas Green

Every day we must make decisions that will affect the future functioning of our "human engine." The choices that precede placing food in one's mouth will help determine what our bodies become. We constantly hear claims made about the benefits and hazards of too much or too little of just about everything. But keeping track of the multitude of nutrients that we need every day is a task that is far too complex and time-consuming without the aid of a computer. This article describes a program that allows the user to "feed" a list of what he ate or would like to eat into a computer. The output obtained provides the quantities of eight items of nutritional interest along with desired totals that are based on the user's age, sex and weight.

The program was written using a

The choices that precede placing food in one's mouth will help determine what our bodies become.

Wang system 2200 that has 8K bytes of memory and uses a cassette tape drive for storing programs and data. It can easily be adapted to any system with similar capacities and can be expanded if a larger system is available. Programs similar to this are already in use in medical centers so that doctors can know exactly what their patients are consuming. [1]

Feeding Your Family On 8K Per Day

With 8K bytes of memory and no disk drives available, the selection of foods must be limited somewhat. It is possible, however, to set up a list of about 100 commonly-consumed foods that will satisfy the needs of most people. There was also a need to limit the number of nutrients that would be analyzed at one time. In this program the author decided to store data regarding eight nutrients for each of 94 food items. The foods selected and the nutrient data are presented in Figure 1. This list is part of the output provided by the program. It is only printed if asked for, however, so as to avoid needless output. This list is an alphanumeric list that is dimensioned by using the statement: DIM A$(100)40. The quantities of each item are listed after the name and are generally given in pounds, cups, teaspoons (TSP), tablespoons (TB), or ounces. For items like apples and oranges, medium size is assumed. The information presented here was obtained from Food Values of Portions Commonly Used, by Bowes and Church. [2] In order to jam all of this data into 8K, it was necessary to store the numerical data as part of an alphanumeric list and convert each entry to a numeric variable just prior to any calculation.

Since your age and sex determine in part how much of a given nutrient you need per kilogram of body weight per day, it is necessary to resort to the use of a number of factors that relate to these variables.

THE AGE AND SEX CATAGORIES ARE:

AGE	SEX	CAT. NO.	AGE	SEX	CAT. NO.	AGE	SEX	CAT. NO.
1	M/F	1	2	M/F	2	3	M/F	3
4-5	M/F	4	6-7	M/F	5	8-9	M/F	6
10-11	M	7	12-13	M	8	14-17	M	9
18+	M	10	10-11	F	11	12-13	F	12
14-17	F	13	18+	F	14			

Figure 2

Although there are more categories than the fourteen listed in Figure 2, everyone can fit into one or another. In some cases categories have been lumped together. Where this is done, the factors used are averages of the factors from food groups that are close enough together so as to cause very little error in the final output. Since the tables you usually see printed are for the man or woman of average weight, the data are not very useful for anyone very far from the mean. Here, however, all of the factors are given per kilogram of body mass so that the dietary needs of a person of any size can be determined. A look at Figure 3 will show how nutrient requirements vary from one group to another.

Drawing The Nutritional Line

It was easy to select calories and protein as important items of interest to most people. Calories are usually of the greatest concern due to the fact that a plus or minus in one's calorie budget will result in a change in weight. It is important information whether you are trying to alter your weight or just maintain a desirable mass. Protein is also of interest, since most of our tissue is composed of protein of some sort. I have yet to run this program for anyone and get less than the required amount of protein, however.

Douglas Green, Cortland Jr.-Sr. High School, Cortland, NY 13045

Diet, cont'd...

The next item on the list is fiber. This is the nondigestible matter that surrounds the cells of plant material. As a result, it acts as nature's own roto-router for the large intestine. Studies have shown that societies where a lot of fiber is consumed have fewer cases of colon cancer than societies like ours where fiber is less prevalent in the diet.[3] Although there is no recommended daily allowance for fiber, most experts in this field feel that fiber is essential for the proper functioning of the digestive tract.

It is possible to set up a list of about 100 commonly consumed foods that will satisfy the needs of most people.

Now for the micronutrients. Here we are concerned with minerals and vitamins. Out of necessity we must narrow the field to representative members. In this program I have chosen calcium, for strong bones and teeth; and iron, to prevent "tired blood." These two seem like the most important minerals around, although there are others you cannot do without. From vitamin land we have vitamin A, for good eyes, vitamin B1, a representative B vitamin, and vitamin C, a favorite of Linus Pauling. Vitamin A is a representative fat-soluble vitamin that comes in foods that tend to contain other such aquaphobes. Foods containing vitamin B1 tend to also be high in other B vitamins hence the term B complex, while vitamin C is a water-soluable vitamin that should be consumed on a daily basis, since we tend to excrete it if it is not used. The units selected were chosen so as to avoid the use of decimals. Thus we have protein listed as grams*10 and vitamin A listed as International units/100. This also conserves space in memory since the decimal point need not be stored, not to mention the fact that many people who like food just fine are scared to death of decimals.

Setting Up The Data File

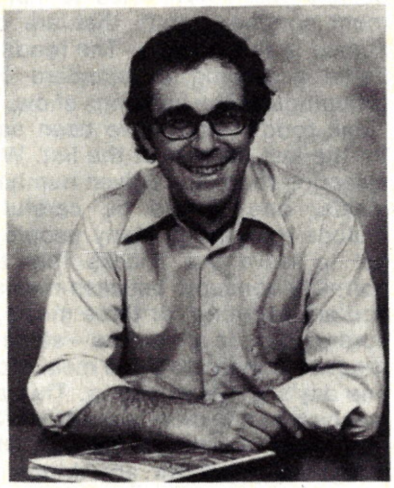
For a system as the one used here it takes a separate program to set up a data file that will be used during the running of the main program. This is accomplished by the use of Program 1. The first set of nested loops allows for the input of the table of factors.

NAME AND QUANTITY	CAL.	PROTEIN	FIBER	CALCIUM	IRON	VIT-B1	VIT-C	VIT-A
1 MEAT-POUL-FISH								
2 BURGER PATTY 1/4	344	292	000	012	040	108	000	001
3 PATTY W/CHEESE	428	341	000	041	041	118	000	001
4 SIRLOIN 1/4LB	424	296	000	012	036	084	000	001
5 PORK 1/4LB	468	296	000	016	040	999	000	000
6 HAM 1/4LB	348	252	000	012	032	560	000	000
7 LIVER 1/4LB	276	316	000	012	104	312	032	640
8 CHICKEN 1/4LB	220	388	000	016	028	080	000	002
9 BOLOGNA 1 SLICE	122	048	000	003	007	064	000	000
10 FRANKFURTER 1	154	062	000	004	009	080	000	000
11 HADDOCK 1/2LB	400	472	000	096	032	096	008	000
12 TUNA 1/4LB	356	348	000	008	024	060	000	001
13 LOBSTER 1/2LB	230	420	000	056	012	999	000	000
14 SHRIMP 1/2LB	276	582	000	276	072	024	000	001
15 BACON 1 SLICE	061	030	000	001	003	051	000	000
16 DAIRY PRODUCTS								
17 MILK WHOLE 1C	159	085	000	288	000	070	002	034
18 MILK SKIM 1C	088	088	000	298	000	100	002	000
19 BUTTER 1TSP	036	000	000	001	000	000	000	017
20 MARGARINE 1TSP	036	000	000	001	000	000	000	017
21 CHEESE CHED 1OZ	120	075	000	225	003	009	000	039
22 COTTAGE CHE.2OZ	051	095	000	053	002	017	000	001
23 EGG 1LG	090	070	000	030	013	060	000	065
24 SALAD DRES 1TB	063	001	000	002	000	000	000	000
25 MAYONNAISE 1TB	108	002	000	003	000	003	000	004
26 CREAM CHEESE 1TB	056	012	000	009	000	003	000	023
27 CREAM LIGHT 2TB	063	009	000	031	000	009	000	003
28 CREAM HEAVY 1TB	053	003	000	011	000	003	000	002
29 BREAD-CEREALS								
30 WHITE 1 SLICE	062	020	000	016	006	060	000	000
31 WHEAT 1 SLICE	056	024	004	023	005	060	000	000
32 BUN-BURGER/FRANK	089	025	001	022	006	080	000	000
33 PANCAKE 1	104	032	001	045	006	080	000	001
34 MUFFIN-CORN	141	032	001	047	008	090	000	001
35 MUFFIN-BLUEBERRY	112	029	001	034	006	060	001	001
36 FRENCH TOAST SLI	183	055	001	077	009	090	000	006
37 WAFFLE 1	209	070	001	085	013	130	000	003
38 ALL BRAN 1/2C	095	031	023	024	029	110	000	000
39 40% BRAN 3/4C	100	028	010	016	013	100	000	000
40 CHEERIOS 1C	102	034	003	042	011	302	000	000
41 CORN FLAKES 1C	095	021	002	006	005	100	000	000
42 OATMEAL 3/4C	098	045	004	153	135	761	000	000
43 RICE KRISPIES 1C	107	016	000	007	005	110	000	000
44 WHEATIES 1C	104	028	005	001	017	167	000	000
45 SHREDDED WHEAT 1	084	022	005	011	008	065	000	000
46 SUGAR CEREAL 1C	140	020	000	000	004	110	000	000
47 MACARONI 1C	207	070	001	015	015	250	000	000
48 RICE 1C	159	032	002	029	012	160	000	000
49 SPAGHETTI 1C	216	073	001	016	016	260	000	000
50 MAC.W/CHEESE 1C	506	189	002	407	020	220	000	010
51 SPAG.W/SAUCE 1C	396	127	007	027	021	120	000	010
52 PIZZA/CHE 1 SLIC	236	120	003	221	010	060	008	006
53 VEGETABLES								
54 BROCCOLI 2/3C	026	031	015	088	008	090	090	025
55 CAULIFLOWER 1C	025	026	011	019	006	045	046	000
56 CELERY 1C RAW	010	005	004	025	003	025	006	001
57 CUCUMBER 1/2C RW	007	003	002	009	002	015	006	000
58 EGGPLANT 1/2C	019	010	009	011	006	050	003	000
59 LETTUCE 4LG LEAV	013	009	005	020	005	060	006	003
60 GREEN PEPPER 1	022	012	014	009	007	080	128	004
61 SPINACH 1/2C	023	030	006	093	022	070	028	081
62 STRING BEANS 1C	025	016	010	050	006	070	012	005
63 SQUASH 1/2C	014	009	006	025	004	050	010	004
64 TOMATO 1 RAW	022	011	005	013	005	060	023	009
65 CARROTS 1/2C RAW	042	012	010	036	008	060	008	110
66 ONIONS 1/2C	029	012	006	024	004	030	007	000
67 PEAS 1/2C	057	043	016	018	014	224	016	004
68 TOMATO JUICE 1/2	019	009	002	007	009	050	016	008
69 COLESLAW 1/2C	099	012	007	043	004	050	029	002
70 TOSSED SALAD	098	020	010	035	011	120	029	012
71 PEANUTS 20 NUTS	114	053	007	015	004	060	000	000
72 PEANUT BUTTER 1T	115	052	004	015	004	024	000	000
73 POTATO-MASH 1/2C	094	021	004	024	004	080	009	002
74 FRENCH FRY 1/4LB	250	041	008	010	020	160	024	000
75 BEANS+PORK 1/4LB	160	083	000	042	022	075	000	002
76 FRUITS								
77 APPLE RAW 1	048	002	008	006	003	024	003	001
78 BANANA RAW 1	042	006	002	004	004	025	005	001
79 ORANGE 1	049	010	005	041	004	100	050	002
80 ORANGE JUICE1/2C	045	007	001	011	002	090	050	002
81 PEACH RAW 1	038	006	006	009	005	020	007	013
82 RAISINS 2TB	043	004	001	009	005	016	000	000
83 PINEAPPLE 1/3C	044	003	001	012	002	040	007	000
84 DESSERTS&MISC								
85 COLA 12 OZ	156	000	000	000	000	000	000	000
86 CHOC MILK 1C	208	068	001	222	005	076	002	003
87 COFFEE BLACK 1C	005	003	000	005	002	010	000	000
88 BEER 12 OZ	160	021	000	015	000	000	000	000
89 BOOZE 1.5 OZ	105	000	000	000	000	000	000	000
90 WINE 4 OZ	090	002	000	000	000	000	000	000
91 ICE CREAM 2/3C	200	034	000	126	000	050	000	004

A REMARKABLE MAGAZINE

creative computing

"The beat covered by Creative Computing is one of the most important, explosive and fast-changing."—Alvin Toffler



David Ahl, Founder and Publisher of Creative Computing

You might think the term "creative computing" is a contradiction. How can something as precise and logical as electronic computing possibly be creative? We think it can be. Consider the way computers are being used to create special effects in movies-image generation, coloring, and computer-driven cameras and props. Or an electronic "sketchpad" for your home computer that adds animation, coloring and shading at your direction. How about a computer simulation of an invasion of killer bees with you trying to find a way of keeping them under control?

Beyond Our Dreams

Computers are not creative per se. But the way in which they are used can be highly creative and imaginative. Five years ago when **Creative Computing** magazine first billed itself as "The Number 1 magazine of computer applications and software," we had no idea how far that would take us. Today, these applications are becoming so broad, so all-encompassing that the computer field will soon include virtually everthing!

In light of this generality, we take "application" to mean whatever *can* be done with computers, *ought* to be done with computers, or *might* be done with computers. That is the meat of **Creative Computing**.

Alvin Toffler, author of *Future Shock* and *The Third Wave* says, "I read **Creative Computing** not only for information about how to make the most of my own equipment but to keep an eye on how the whole field is emerging."

Creative Computing, the company as well as the magazine, is uniquely lighthearted but also seriously interested in all aspects of computing. Ours is the magazine of software, graphics, games and simulations for beginners and relaxing professionals. We try to present the new and important ideas of the field in a way that a 14-year old or a Cobol programmer can understand them. Things like text editing, social simulations, control of household devices, animation and graphics, and communications networks.

Understandable Yet Challenging

As the premier magazine for beginners, it is our solemn responsibility to make what we publish comprehensible to the newcomer. That does not mean easy; our readers like to be challenged. It means providing the reader who has no preparation with every possible means to seize the subject matter and make it his own.

However, we don't want the experts in our audience to be bored. So we try to publish articles of interest to beginners and experts at the same time. Ideally, we would like every piece to have instructional or informative content—and some depth—even when communicated humorously or playfully. Thus, our favorite kind of piece is accessible to the beginner, theoretically non-trivial, interesting on more than one level, and perhaps even humorous.

David Gerrold of *Star Trek* fame says, "**Creative Computing** with its unpretentious, down-to-earth lucidity encourages the computer user to have fun. **Creative Computing** makes it possible for me to learn basic programming skills and use the computer better than any other source."

Hard-hitting Evaluations

At **Creative Computing** we obtain new computer systems, peripherals, and software as soon as they are announced. We put them through their paces in our Software Development Center and also in the environment for which they are intended-home, business, laboratory, or school.

Our evaluations are unbiased and accurate. We compared word processing printers and found two losers among highly promoted makes. Conversely, we found one computer had far more than its advertised capability. Of 16 educational packages, only seven offered solid learning value.

When we say unbiased reviews we mean it. More than once, our honesty has cost us an advertiser—temporarily. But we feel that our first obligation is to our readers and that editorial excellence and integrity are our highest goals.

Karl Zinn at the University of Michigan feels we are meeting these goals when he writes, "**Creative Computing** consistently provides value in articles, product reviews and systems comparisons. . .in a magazine that is fun to read."

Order Today

To order your subscription to **Creative Computing**, send $15 for one year (12 issues), $28 for two years (24 issues) or $40 for three years (36 issues). If you prefer, call our toll-free number, 800-631-8112 (in NJ 201-540-0445) to put your subscription on your MasterCard, Visa or American Express card. Canadian and other foreign surface subscriptions are $24 per year, and must be prepaid. We guarantee your satisfaction or we will refund the unfulfilled portion of your subscription.

Join over 80,000 subscribers like Ann Lewin, Director of the Capital Children's Museum who says, "I am very much impressed with **Creative Computing.** It is helping to demystify the computer. Its articles are helpful, humorous and humane. The world needs **Creative Computing**."

creative computing

P.O. Box 789-M
Morristown, NJ 07960
Toll-free **800-631-8112**
(In NJ 201-540-0445)

Diet, cont'd...

NAME AND QUANTITY	CAL.	PROTEIN	FIBER	CALCIUM	IRON	VIT-B1	VIT-C	VIT-A
92 DOUGHNUT 1	125	015	000	013	004	050	000	000
93 CAKE/ICED 1 SL	208	020	000	046	008	011	000	001
94 COOKIE CHOC CHIP	051	006	000	004	002	004	000	000
95 GELATIN 2/3C	109	022	000	003	000	009	010	000
96 POTATO CHIPS 10	108	014	002	003	002	018	001	000
97 PIE APPLE 1/6PIE	410	034	006	001	005	030	002	001
98 SUGAR 1TSP	016	000	000	000	000	000	000	000
99 CATSUP 1TB	019	003	000	004	001	020	003	002
100 FRUIT DRINK 1C	106	000	000	000	000	000	003	000

Figure 1

This 14 by 7 array shown in Figure 3 is stored as matrix F( ). While these numbers are being input, the program continually prints the position of the factor in the matrix on the screen so that the operator can keep track of which piece of data is to be entered next. The next loop allows for the input of the list of foods along with the nutrient profile of each. Included in this list are six headings. These are used to help organize the list and to make the foods easy to find on the printed list (Figure 1). The name and quantity of each food item must fit into the first 16 spaces of each member of the list A$( ). For this reason it is easier to input the name and quantity as a separate entry. This saves the trouble of counting the number of spaces used, so that the number of calories begins in the 17th column. The remaining 24 spaces in each variable are filled with the nutritional data for each item except of course for the headings. The data for each of the eight nutrients is allotted three spaces, and zeros are used to fill in any spaces that remain on the left. This must be done so that the computer does not find itself trying to convert a space into a number.

After the data have all been input, the remaining statements in Program 1 serve to save it on your tape. Be sure to check the user's manual for your system in order to obtain the correct statements. Here, the DATA SAVE OPEN "FOOD" places a leader on the tape, so that the beginning of the data file is marked. DATA SAVE A$( ), F( ) causes the computer to record the values from lists A$ and F currently in the memory onto the tape, while DATA SAVE END marks the end of the file. The DATA LOAD statements at the beginning of Program 2, the main program, cause the reverse process to take place. Here the information is loaded off the tape and into the computer's memory. What, no Brie?

The tables you usually see are for the average; here, however, the dietary needs of a person of any size can be determined.

It seems that everyone who looks at the list in Figure 1 tries immediately to find their favorite exotic food that is certain not to be listed. Having only room for 94 items, I had to make some decisions as to what should be included and what would be left out. I have tried to list the most common foods that are consumed in this society. The foods are listed according to standard food groupings as the headings show. The "junk food" items have been saved for the last section of the list. When selecting foods for a given number of meals, one should be careful to include everything. Many people are apt to leave out snacks that they normally consume between meals. If you are planning for three meals, be sure to include all snacks—since the program considers three meals to be one day's worth of eating. This also means that one meal is exactly 1/3 of one day's worth of eating. Other things to check for are things like hamburgers and hotdogs, which require also the addition of the roll. (This adjustment allows the user to eat just plain hamburgers or hot dogs without the roll if desired.) For convenience, however, tossed salad is listed as a separate item to save the user the trouble of tossing their own.

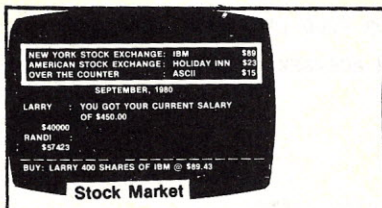
If you are serious about using this program, but find the selection of foods to be a problem, you can simply scan the list and find food items that you never eat and substitute the items you feel should have been there. Another option is to replace the headings with foods that you eat often that do not appear on the list. When adding your own food items, be sure to get the correct units for each of the nutrients. A good way

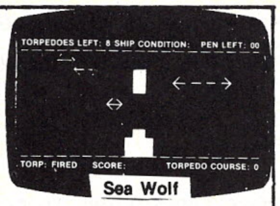
FACTORS USED TO DETERMINE DESIRED TOTALS

CATAGORY	CALORIES	PROTEIN GM*10	CA MG	IRON MG*10	B1 MCG	C MG	A IU/100
1	91.7	20.8	58.3	12.5	50.0	3.33	1.67
2	89.3	17.9	57.1	10.7	42.9	2.86	1.43
3	87.5	18.8	50.0	6.2	43.8	2.50	1.56
4	84.2	15.8	42.1	5.2	42.1	2.11	1.32
5	87.0	15.2	39.1	4.3	43.5	1.74	1.52
6	78.6	14.3	35.7	3.5	39.3	1.43	1.25
7	32.5	12.9	34.3	2.8	37.1	1.14	1.28
8	62.8	11.6	32.6	4.1	32.6	1.05	1.16
9	50.8	10.2	23.7	3.0	25.4	0.93	0.84
10	41.8	9.0	11.2	1.4	20.9	0.90	0.74
11	64.3	14.3	34.2	5.1	31.4	1.14	1.29
12	52.3	11.4	29.5	4.1	27.3	1.02	1.14
13	44.3	10.3	24.5	3.4	22.7	0.94	0.94
14	34.5	9.5	13.8	3.1	17.2	0.94	0.86

NOTE: ALL FACTORS ARE PER KG PER DAY

Figure 3





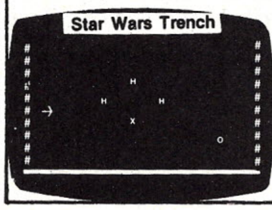
NEW! ASCII
for the TRS-80*

Every month you receive a certified ASCII C-20 cassette containing: a cover page with a directory of programs
4 original programs
an information packed newsletter
and information on ASCII funded contests

Rates:	1 year [12 issues]	$35.00
	6 months [6 issues]	$20.00
	Sample issue	$ 4.00

Write For Overseas Rates
To subscribe, write to: ASCII
P.O. Box 516, Valley Stream, N.Y. 11582
Or call: (515) 791-4723

Level II16K Required

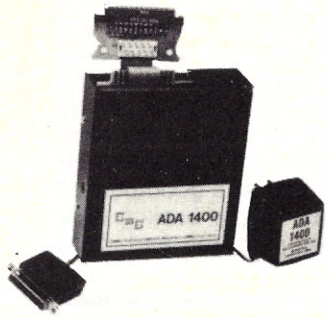


***TRS-80 is a product of Tandy Corp.**

Makes A Great Gift

CIRCLE 116 ON READER SERVICE CARD

PET Printer Adapter



CmC's ADA 1400 drives a printer with an RS-232 interface from the Commodore PET IEEE-488 bus. The ADA 1400 is addressable, works with the Commodore disk and prints upper and lower case ASCII.

A PET IEEE type port is provided for daisy-chaining other devices.

A cassette tape is included with programs for plot routines, data formatting and screen dumps. The ADA 1400 sells for $179.00 and includes a PET IEEE cable, RS-232 cable, power supply, case, instructions and software.

Order direct or contact your local computer store.

VISA & M/C ACCEPTED-SEND ACCOUNT NO., EXPIRATION DATE AND SIGN ORDER.
ADD $3 PER ORDER FOR SHIPPING & HANDLING-FOREIGN ORDERS ADD 10% FOR AIR POSTAGE.

CmC **Connecticut microComputer, Inc.**
34 Del Mar Drive, Brookfield, CT 06804
203 775-4595 TWX: 710 456-0052

NEW EATON 7000+ PERSONAL COMPUTER PRINTER

PLUG IN AND PRINT OUT— IN SECONDS!

Now you can add hard copy capability to your computer with plug-in simplicity that matches up to any micro system. And with the Eaton 7000+ dot-matrix printer, you get complete printout versatility at an affordable price.

Interfaces with any personal computer
Apple, Commodore Pet, TRS-80, Northstar and others—no matter what type of computer you have or are thinking of buying, Eaton 7000+ fits. Just plug it in and start printing. You'll get virtually maintenance-free operation with a minimum of moving parts. And minimum life of 100 million characters with Eaton's newest printhead, while the print mechanism is designed for a minimum of 10 million cycles.

High performance design
The 7000+ features uni-directional printing at a speed of 1.25 lines per second. It accepts any standard single or 2-ply roll paper from ¾" to 3⅞" wide and prints 3⅓" line with 40 to 64 adjustable character capacity.

For immediate information call: Eaton Printer Product Operations, 307/856-4821.

For descriptive literature and the name of your nearest dealer write Eaton Corporation, Count Control/Systems Division, 901 South 12th St., Watertown, WI 53094.

Dealer inquiries invited.

EATON
Electrical/Electronic Control

CIRCLE 153 ON READER SERVICE CARD EAT-1A

Diet, cont'd...

to check this out is to see if your values appear to be in line with the other values on the list. If any of the numbers seem too high or too low, you may have your decimal in the wrong location. This is one of the ways I checked the original list after it had been input. Since entering all of this data represents a good deal of work, be sure to keep a backup tape with the same program and data files.

Many people are apt to leave out snacks that they normally consume between meals.

How The Program Works

A look at the flowchart in Figure 4 will provide an overview of how the program works. After the program is loaded into memory, the data file should then be positioned so that it will be the next in line. If the data is on a separate tape, then the program tape must first be removed and replaced with the data tape. After the command run is executed, the computer will load the data into the space saved for the food list, A$( ) and the factor table F( ). This is accomplished by data load statements 50 and 60. With the data in place the user is then asked if a list of the foods is needed. If this is the case, a list is provided by statements 440 through 540. The name of each item is printed as part of an outer loop that goes from 1 to 100. An inner loop from 1 to 8 allows for the printing of the eight nutrients without the extra fuss of a printusing statement. (In order to do this sort of thing it is necessary to use a statement like number 500 where the value of the TAB is a function of the loop counter. The ability to figure out such functions by the "seat of the pants" is a skill that any professional programmer must possess.)

Now we come to the inputs that describe the user. This requires that the list of age/sex categories shown in Figure 2 be displayed on the CRT. In addition to selecting the category the user must also include his weight in pounds. The category number tells the computer which row of factors to use when determining the desired totals. The weight is converted into kilograms and multiplied by each factor in order to arrive at the daily required values for the operator. With the user characteristics in place, it is time to enter the numbers of the items in the user's menu along with the number of units of each item and the number of meals that will be represented by the food selected. As these menu items are entered, a list of the foods selected is printed so that the total menu can be surveyed after it has been fed in. As the items are entered,the computer maintains a list of totals for each of the eight nutrients. This is the list Q(). This is

```
10 REM ***SAVE ROOM IN MEMORY FOR FOOD LIST AND FACTORS***
20 DIM A$(100)40,F(14,7)
30 REM ***INPUT FACTORS FOR EACH AGE/SEX GROUP***
40 FOR N = 1 TO 14
50 FOR M = 1 TO 7
60 PRINT N;M
70 INPUT F(N,M)
80 NEXT M
90 NEXT N
100 STOP "MAKE CORRECTIONS TO FACTOR LIST"
110 REM ***INPUT VALUES FOR FOOD LIST A$()***
120 FOR  N = 1 TO 100
130 REM ***INPUT NAME AND QUANTITY***
140 PRINT N
150 INPUT STR(A$(N),1,16)
160 REM ***INPUT NUTRIENT DATA***
170 INPUT STR(A$(N),17,24)
180 NEXT N
190 STOP "MAKE CORRECTIONS TO FOOD LIST"
200 REM ***SAVE DATA ON TAPE***
210 DATA SAVE OPEN "FOODS"
220 DATA SAVE A$(),F()
230 DATA SAVE END
240 END
```

Program #1

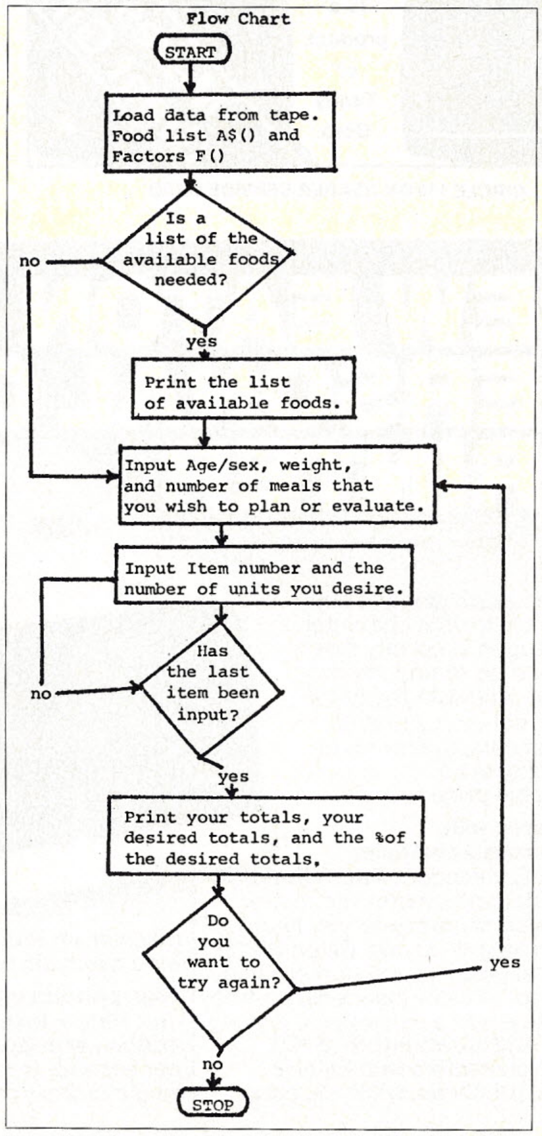


Figure 4

Diet, cont'd...

done in statements 340 to 380 with the aid of a loop from 1 to 8 and the CONVERT statement. Here the data from the alphanumeric list is picked out three figures at a time and converted to numbers so that the numeric operation of addition can be performed. Note also that the starting location of the data selected from the string is a function of the loop counter N. This is similar to the use made of the loop counter in the TAB statement mentioned earlier.

With the input of the number zero the system begins the task of providing output. The samples in Figure 5 illustrate how this is done. The row labeled your totals is simply the data stored in list Q(). The desired totals are arrived at by multiplying the appropriate factors

```
10 REM ***REM STATEMENTS CAN NOT BE USED ON A SYSTEM WITH ONLY 8K OF MEMORY***
20 REM ***SAVES SPACE FOR FACTORS(F), LIST OF FOODS(A$), AND YOUR TOTALS(Q)***
30 DIM F(14,7),A$(100)40,Q(8)
40 REM ***LOADS THE LIST OF FOODS AND FACTORS OFF OF A TAPE***
50 DATA LOAD "FOOD"
60 DATA LOAD A$(),F()
70 REM ***ASKS IF YOU WANT A LIST OF FOODS AVAILABLE***
80 INPUT "DO YOU NEED A LIST OF THE AVAILABLE FOODS(Y OR N)",Y$
90 IF Y$="Y" THEN 430
100 REM ***HERE IS WHERE THE AGE AND SEX CATEGORIES ARE PRINTED***
110 PRINT "THE AGE AND SEX CATAGORIES ARE:"
120 PRINT
130 PRINT "AGE   SEX   CAT. NO.    AGE   SEX   CAT. NO.    AGE   SEX   CAT. NO."
140 PRINT " 1    M/F     1          2    M/F     2           3    M/F     3"
150 PRINT "4-5   M/F     4         6-7   M/F     5          8-9   M/F     6"
160 PRINT "10-11 M       7        12-13   M      8        14-17   M       9"
170 PRINT "18+   M      10        10-11   F     11        12-13   F      12"
180 PRINT "14-17 F      13         18+    F     14"
190 REM ***ASKS FOR CHARACTERISTICS OF OPERATOR***
200 INPUT "ENTER THE NUMBER OF YOUR CATAGORY",C
210 INPUT "WHAT IS YOUR WEIGHT IN POUNDS",W
220 INPUT "HOW MANY MEALS DO YOU WANT TO PLAN OR EVALUATE",P
230 PRINT "ENTER THE NUMBER AND QUANTITY OF EACH FOOD YOU WISH TO EAT"
240 PRINT "FOR THE NUMBER OF MEALS YOU HAVE SELECTED.  ENTER A 0 WHEN YOU ARE FINISHED."
250 REM ***SENDS OUTPUT TO THE PRINTER***
260 SELECT PRINT 215(90)
270 REM ***BEGINS TO PRINT A LIST OF FOODS SELECTED***
280 PRINT "NO.";"  ITEM  ";TAB(18);"QUANTITY"
290 REM ***ITEM NUMBER AND QUANTITY INPUT HERE***
300 INPUT "ITEM NUMBER",I
310 REM ***THIS IS A FLAG THAT SENDS YOU TO OUTPUT SECTION WHEN INPUT IS FINISHED***
320 IF I=0 THEN 570
330 INPUT "NUMBER OF UNITS OF THIS ITEM",U
340 REM ***HERE IS WHERE YOUR TOTALS ARE COMPILED***
350 FOR N=1TO 8
360 CONVERT STR(A$(I),N*3+14,3) TO X
370 Q(N)=Q(N)+X*U
380 NEXT N
390 REM ***THIS PRINTS THE ITEM YOU JUST SELECTED***
400 PRINT I;STR(A$(I),1,16);TAB(22);U
410 GOTO 300
420 REM ***SENDS OUTPUT TO PRINTER***
430 SELECT PRINT 215(90)
440 REM ***THESE STATEMENTS PRINT THE LIST OF FOOD (A$)***
450 PRINTUSING 460
460 %NAME AND QUANTITY    CAL.  PROTEIN  FIBER  CALCIUM  IRON    VIT-B1  VIT-C   VIT-A
470 FOR N=1TO 100
480 PRINT N;STR(A$(N),1,16);
490 FOR M=1TO 8
500 PRINT TAB(21+8*(M-1));STR(A$(N),M*3+14,3);
510 NEXT M
520 PRINT
530 NEXT N
540 REM ***THIS RETURNS OUTPUT TO CRT***
550 SELECT PRINT 005(64)
560 GOTO 110
570 REM ***THIS IS THE BEGINNING OF YOUR OUTPUT***
580 PRINTUSING 590
590 %                     CAL.  PROTEIN  FIBER  CALCIUM  IRON    VIT-B1  VIT-C   VIT-A
600 PRINTUSING 610,Q(1),Q(2),Q(3),Q(4),Q(5),Q(6),Q(7),Q(8)
610 %YOUR TOTALS          #####  #####   #####  #####    #####   #####   #####   #####
620 PRINT
630 REM ***FUNCTION 1 MULTIPIES A FACTOR BY YOUR WEIGHT IN KG AND THE NUMBER OF DAYS YOU ARE E
ATING FOR***
640 DEFFN1(X)=INT(P/3*W/2.2*X+.5)
650 PRINTUSING 660,FN1(F(C,1)),FN1(F(C,2)),FN1(F(C,3)),FN1(F(C,4)),FN1(F(C,5)),FN1(F(C,6)),FN1
(F(C,7))
660 %DESIRED TOTALS       #####  #####    ???   #####    #####   #####   #####   #####
670 PRINT
680 REM ***FUNCTION 2 CONVERTS YOUR TOTALS INTO A % OF THE DESIRED TOTALS***
690 DEFFN2(X)=INT(X*100+.5)
700 PRINTUSING 710,FN2(Q(1)/FN1(F(C,1))),FN2(Q(2)/FN1(F(C,2))),FN2(Q(4)/FN1(F(C,3))),FN2(Q(5)/
FN1(F(C,4))),FN2(Q(6)/FN1(F(C,5))),FN2(Q(7)/FN1(F(C,6))),FN2(Q(8)/FN1(F(C,7)))
710 %PER CENT OF DESIRED ####   ####     ???    ####     ####    ####    ####    ####
720 INPUT "DO YOU WANT TO TRY AGAIN(Y OR N)",Y$
730 REM ***THIS SETS YOUR TOTALS EQUAL TO ZERO***
740 MAT Q = ZER
750 REM ***THIS RETURNS OUTPUT TO CRT***
760 SELECT PRINT 005(64)
770 IF Y$="Y" THEN 200
780 END
```

Program #2 The Main Program

Diet, cont'd...

by the body weight and the number of days worth of meals being evaluated. Special function #1 defined in statement 640 is used for this reason. It is important to note here that there is no recommended daily allotment for fiber. This is the reason for the question marks that appear in the output. The percent of desired totals row is obtained by dividing the user's totals by the desired totals and converting the result to a percent. This is done with the help of special function #2 that is defined in statement 690. With the output in hand, the opportunity is presented to try again. If the answer is yes, control is shifted to statement 200 which requires that all of the user characteristics be input again. This facilitates the program for use by several different people at the same time as would be desirable in a classroom setting. For home use, statement 770 might be changed to read IF Y$ = "Y" THEN 240. This foregoes the need to reenter the user's characteristics.

Reading The Nutritional Report Card

In Figure 5 you can see what two samples of output look like. The first is a rather typical day's selection for the author, while the second is your basic "fast food" meal. Although fast-food milkshakes do not usually contain milk, using milk and ice cream in the place of a shake does not lead to great error. From this sample output it is easy to see that the author practices what he preaches, and that the typical fast food meal is loaded with calories and protein but shorter on some of the other basics. The ideal output would have the percentages of all other nutrients greater than the percentage for calories. Otherwise you could be "full," but not full of the proper nutrients.

Such analysis allows the user to make corrections at the next meal for indulgences committed at a previous one. It also allows one to spot dietary deficiencies for critical vitamins and minerals, and to keep a close count of the personal calorie budget.

For people who exercise, the desired number of calories may be too low. If you are a jogger, you can figure an extra 100 calories or so per mile traveled.

For people who exercise, the desired number of calories may be too low. If you are a jogger, as more and more of us are nowadays, you can figure an extra 100 calories or so per mile traveled, regardless of the speed at which you plod along. If, however, you are exceptionally sedentary, then you may find the calorie count higher than you would require to maintain a stable weight. Also, do not forget those African tribesmen who eat tons of fiber and have very little cancer of the large intestine. If your fiber intake is too low then you may be asking for trouble, not to mention irregularity. For pill-poppers, the output shows if vitamin supplements are really needed.

Possible Modifications

If you are getting past middle age or if you are taking medication to control your blood pressure, then the nutrients listed here may not be the most important ones for you. In addition to calories you would probably rather keep track of things like sodium, cholesterol and triglycerides. You may even wish to differentiate between saturated and unsaturated fats. This will require that you go to a source like the one listed in this article and obtain the necessary information for each of the nutrients that are important to your own personal health. If you really want to get technical, and memory space is no problem, you can even keep track of the amino-acid profiles for each of your protein-bearing items. This is being done in some hospitals today, and it is one way for vegetarians to be sure that they are getting the correct mixture of vegetable proteins. If you have gone to the trouble of setting up a program such

```
NO.  ITEM          QUANTITY
 8 CHICKEN 1/4LB        1
 12 TUNA 1/4LB          1
 17 MILK WHOLE 1C       2
 18 MILK SKIM 1C        1
 20 MARGARINE 1TSP      6
 25 MAYONNAISE 1TB      2
 31 WHEAT 1 SLICE       6
 45 SHREDDED WHEAT 1    2
 56 CELERY 1C RAW       1
 70 TOSSED SALAD        2
 73 POTATO-MASH 1/2C    2
 77 APPLE RAW 1         1
 80 ORANGE JUICE1/2C    1
 87 COFFEE BLACK 1C     3
 88 BEER 12 OZ          2
 94 COOKIE CHOC CHIP    3
                     CAL.   PROTEIN   FIBER   CALCIUM   IRON    VIT-B1   VIT-C   VIT-A
YOUR TOTALS          2893    1351      75      1287      148     1457     141     213

DESIRED TOTALS       2850     614      ???      811      102     1425      61      51

PER CENT OF DESIRED   102     220      ???      159      145      102     231     418


NO.  ITEM          QUANTITY
 2 BURGER PATTY 1/4     2
 17 MILK WHOLE 1C       1
 32 BUN-BURGER/FRANK    2
 74 FRENCH FRY 1/4LB    1
 91 ICE CREAM 2/3C      1
                     CAL.   PROTEIN   FIBER   CALCIUM   IRON    VIT-B1   VIT-C   VIT-A
YOUR TOTALS          1475     794      10       492      112      656      26      40

DESIRED TOTALS        950     205      ???      270       34      475      20      17

PER CENT OF DESIRED   155     387      ???      182      329      138     130     235
```

Figure 5

Diet, cont'd...

as this, you may even wish to advertise your services to people who are interested in this type of information but who for some reason do not have access to this type of evaluative tool. This would also be an easy way to evaluate the foods served in institutions to make sure that they are up to federal regulations. Perhaps the dietician in your local high school could use this type of help to liven up the academic bill of fare. I have tried adding cost as a factor, but with seasonal items, specials and inflation this has not proved practical for anything other than an educational simulation. Further possibilities are limited only by the user's imagination, due to the fact that eating is one habit we all must indulge in for as long as the human engine is left on.

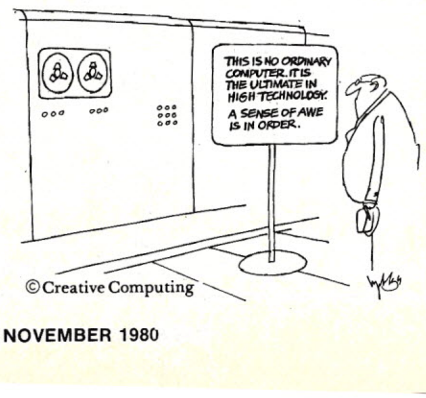
References

1. IBM advertisement, *Datamation* Oct. 79, p. 78.
2. Bowes and Church, *Food Values of Portions Commonly Used*, 11th ed. (J.B. Lippincott and Co., Philadelphia, 1970).
3. James Scala, *The Physiological Effects of Dietary Fiber*, ACS Symposium Series #15, American Chemical Society, Washington D.C., 1975), pp. 325-335.

Tape and Disk Version for TRS-80

A different version of this program, also called Diet, is available for the TRS-80 (Level II, 16K) from Creative Computing Software. It analyzes four groups of basic substances: protein, lipids, calories and carbohydrates instead of the seven mentioned in the article. It is "menu" driven and screen oriented and allows for up to four meals per day. It indicates the effect of a daily diet on body weight and nutrition.

Three other programs are on the same tape (or disk): Pollute, Rats and Malaria. Order "Ecology Simulations—2," CS-3202 for cassette tape or CS-3502 for floppy disk. Price is $25.95 postpaid for either one from Creative Computing Software, P.O. Box 789-M, Morristown, NJ 07960. □



TRcopy and the Pirates

Stephen Kimmel

Cassette storage of data and programs has always been a big problem. I don't know how many cassette programs I've purchased that were difficult or impossible to load. I've lost indexes and couldn't remember what was on which tape. I've had mechanical failures of the cassette *boxes*. My kid even chewed a hole in one tape, destroying Wumpus. Then there was the frustration of loading a program into the machine, with everything apparently all right, little stars flashing and everything, only to discover that what I had was a whole lot of garbage. Machine-language tapes are the worst of the lot. I could never figure out how to make backup copies so that I wouldn't be out $20 if there was a bolt of lightning somewhere in the state.

TRcopy seemed too good to be true. It seemed like a godsend.

TRcopy is a machine language program for the TRS-80 by Data/Print Publishing of Fargo, N.D. Their phone number is 1-800-437-4144. This program displays the information coming off the cassette as it comes off the tape, instantly telling you whether you are getting a good load or not. You can also tell which tape it is you are loading in. That's only the beginning of what TRcopy does.

After loading the program into the computer, you can verify that the computer understands the tape the same way twice. This feature checks the data, byte for byte, against the program stored in memory. Then it tells you whether the two match with an appropriate good or bad message.

Having verified that the program is in memory correctly, you can then copy it back onto another tape and verify it against the memory. If the new tape and the old tape both verify against the same memory contents, then they must be identical. And they are. Except that the new tape is probably better than the original, having been subjected to fewer magnetic fluxes and whatever.

That the programmer of TRcopy felt the need to build in a self-defense mechanism says something very interesting about the program . . . and about the state of the software industry in general.

TRcopy is easy to use. Their instruction book is clear and easy to understand and has a lot of other useful information in it. And all the Data/Print folks want for TRcopy is (gasp!) $39.95. It does everything they said it would do, but $39.95?

There is a problem. The program loaded in under TRcopy can't be executed. You have to dump TRcopy and load the program in again to execute it.

What I wanted was a program that would give me visual confirmation of a good load and then allow me to run the program. TRcopy wasn't worth the money to me so I sent it back for my refund. (There are a number of similar programs — Duplik, SYSCOP, COPSYS, Clone and others. I think the best dollar value is probably Duplik, which comes with a Renumber program for $8.)

Musings

The instruction book stated that TRcopy couldn't be used to copy itself. Hmmmm. Of course I tried and they're right. If you try, the program automatically dumps itself. I'm sure some machine language whiz could figure some means of defeating the defense mechanism but it was beyond me.

That the programmer of TRcopy felt the need to build in a self-defense mechanism says something very interesting about the program . . . and about the state of the software industry in general. Let's face it. TRcopy and the others, Duplik, SYSCOP, COPSYS, Clone, etc. are the computer world's software answer to the Xerox machine. They are programs designed to violate copyright laws.

Everyone knows that the cost of software is high. You could easily spend

Stephen Kimmel, 4756 S. Irvington Place, Tulsa, OK 74135.

four times the cost of the hardware on computer programs that are only mediocre.

With TRcopy, it is possible to pool your resources. Ten people, say, chip in $2 apiece and buy one program. Then they take their handy dandy duplicating program and make nine copies. Everyone has a perfectly good, perhaps even better, copy of the original. Let's go whole hog. Let's make ten copies, send the original back and demand our money back. *And* since we still have TRcopy and the program we can make ten more copies and sell them. Or a hundred. Or a thousand. Suddenly there are a zillion copies of the $20 program running around. They aren't cheap inferior copies. They are cheap *superior* copies. They are also highly illegal.

I decided to try a similar tactic. I took my TRcopy to a local Radio Shack. With the manager's help, I managed to copy

With the manager's help, I managed to copy nearly $200 worth of software onto a $2 cassette tape inside of twenty minutes.

nearly $200 worth of software onto a $2 cassette tape inside of twenty minutes. TRcopy, thus, becomes an interesting means of shoplifting. It is also curious in that the local Radio Shack isn't out anything. They've made a $2 sale, where they probably wouldn't have made the $200 sale.

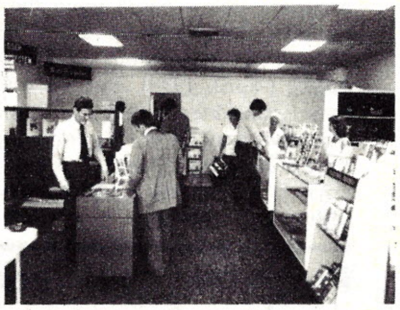
Perhaps TRcopy should be illegal. Still, the program is irrelevant. (The Xerox machine isn't the counterfeiter.) The duplicating programs are inevitable and the fact it can be done means that it will be done. TRcopy is only the most expensive of the group.

It is a mark of the maturity of our industry that we have finally produced our own pirate industry. Have we developed a group of programs capable of destroying the fabric of the software industry? Perhaps. The marginal houses with one terrific program will suffer the most. Unfortunately, the better their program, the more they'll suffer. The one sure way to eliminate this sort of thing is to make software so cheap that there is no incentive to stealing it. But that would leave no incentive to develop software worth stealing or buying.

Remember the old Terry and the Pirates comic strip? No matter how strange the machinations, the forces of good always triumphed eventually. With our current pirate situation, I have a feeling the forces of good will come out on the short end this time. □

Computer Store of the Month

Hundreds of computer stores sell *Creative Computing* Magazine, Press books and software nationwide. We believe that the contributions made by these stores to their communities should be recognized and applauded. In this and future issues of *Creative Computing* Magazine, we'll spotlight some stores which deserve attention for their salesmanship, creativity and community service.



Home Computer Center Inc. opened it's first store in July 1977 in Virginia Beach, Va and opened a second shortly after in Newport News, Va. Both outlets experienced fast growth by offering a wide selection of personal computers, a full line of personal computer software, complete libraries of computer literature, and support.

Home Computer Center Inc. has expanded into several different areas. The first includes a full line of terminals and communication equipment to meet the needs of local business; additionally, they have established a division which will handle products for the education field. They have also started to produce their micro-hardware including a calendar clock and the Apple Butler for use with the Apple Computer.

Home Computer Center sells *Creative Computing* magazine, books and software. If you are in Virginia you may want to look up one of the outlets located at 2927 Virginia Beach Blvd., Virginia Beach or 12588 Worwich Blvd., Newport News. You will find the staff knowledgable, helpful and friendly.

The Shairs, owners of The Computer Corner of White Plains, the featured store of the month in July, 1980, sent this announcement. Congratulations, Shairs.

Order No. 3

Hal & Harriet Shair
41 COLBY AVENUE
RYE, NEW YORK

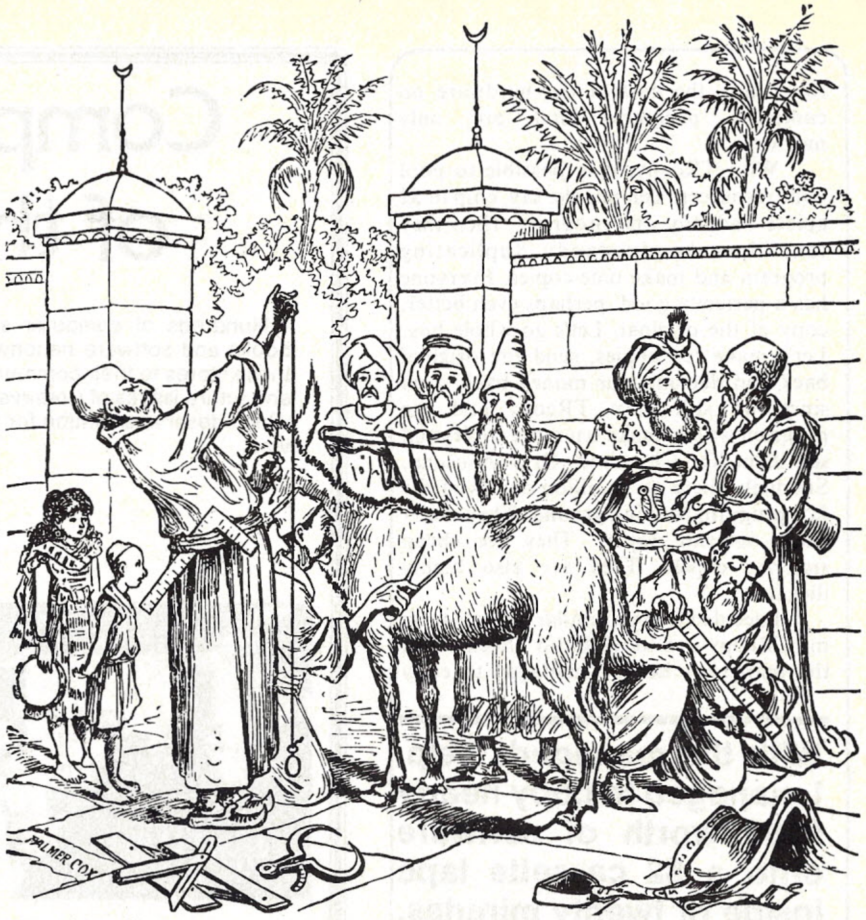
GREENWICH HOSPITAL
Perryridge Road
Greenwich, Connecticut

Department: Kiddies
Delivery Date: August 29, 1980
F.O.B.: Greenwich, Conn.
Terms: Net 9 months
Via: Stork Express

MODEL	QUANTITY	DESCRIPTION AND COLOR	SIZE	PRICE
DAVID RYAN SHAIR	(1)	Healthy Infant (Male) Rosy Complexion Capable of Loud Noises SHIPMENT RECEIVED IN GOOD CONDITION at 4:56 P.M.	6 lbs. 10 oz. 20 inches	Priceless HAL & HARRIET BUYERS

This is to certify that the model described herein was produced in accordance with the Fair Labor Laws applicable thereto

Base Arithmetic



David B. Suits

I sometimes need to convert numbers from one base to another — usually binary or hex to decimal, or else the other way around. Sometimes I need to know the binary equivalent of a hex number or *vice-versa.* Programs to convert numbers from one base to another are not difficult to write, but they are limited.

Get out your base conversion program and find the decimal equivalent of 3FCA(hex). No problem, you say? Okay, then. What is the decimal equivalent of $17F^4$(hex) + 110011(binary)? What is the octal equivalent of 15(hex) * 256(decimal) - 1101101(binary)? If your base conversion program is anything like my *old* one, you probably had to use pencil and paper — and maybe a calculator as well — on those last two problems. Perhaps you converted all the numbers to some common base (probably decimal), performed the necessary arithmetic, and then converted the result to the desired base. But it is silly to waste your time on each of these steps if your computer can do it all for you.

The "Base Arithmetic" program will perform addition, subtraction, multiplication and division of integer values (positive or negative) on simple or complicated expressions (where the numbers may be expressed in binary, octal, decimal or hexadecimal), and print out the result in the base of your choice. (For the purposes of formatting the output, the expression must evaluate to a number between -65535 and +65535, inclusive.)

David B. Suits, General Studies, Rochester Institute of Technology, Rochester, NY 14623.

The idea for the parser which this program uses is based upon the "people parser" described by J. W. Garson, "People Programming," *Creative Computing* (April, 1979). But while Garson's parser parsed from right to left, mine goes the other way. It is very forgiving about mismatched parentheses, and if a fatal error is found, the program tries to be informative about it.

Programs to convert numbers from one base to another are not difficult to write, but they are limited.

The program uses two arrays to simulate two first-in-last-out stacks — a number stack NS( ) and an operator stack OS$( ) — plus a stack pointer for each: NP and OP. Each time the parser encounters a number in the input expression, it converts it to decimal, pushes it onto the number stack and increments the number stack pointer. If an operator is found, it decides whether it is a binary operator (+, -, *, or /) or a unary operator (+ or -). If it is a unary operator is is applied to the number which follows. But if it is a binary operator (or a parenthesis) it is either pushed onto the operator stack or else applied to the top two numbers on the number stack. Whether the operator is pushed onto the stack or applied to the numbers on the number stack is determined by the "strength" of the operator as compared to the strength of the operator (if any) at the top of the operator stack. Following Garson, the operator strengths are as given in lines 72-80 of the program listing. (The right parenthesis is not included in the table of strengths because, should it be encountered in the input expression, the operator on top of the stack is immediately applied to the top two numbers on the number stack.) These values are held in the array S(1) through S(7), where the indices are calculated by subtracting 40 from the ASCII value of the operator. If the strength of the operator discovered in the input expression is *less* than the operator (if there is one) on top of the operator stack, then the top two numbers on the number stack are popped off the stack, the operator on the operator stack is popped off and applied to the numbers, and the result is pushed back onto the number stack. The procedure is repeated until the strength of the operator from the input expression is not less than the strength of the operator on top of the operator stack, in which case the new operator is pushed onto the operator stack.

During the parsing process, the buffer pointer (BP) keeps track of the position of the present character being examined in the input expression, I$. The value of BP thus allows an error message to point to the specific place in the input expression where an error is first encountered.

The parser can be defeated, if you try. It is not fully idiot proof.

Necessary run time instructions are given in the listing.

Arithmetic, cont'd...

```
READY

LOAD"CONVRT":RUN30
BASE ARITHMETIC   BY DAVID B. SUITS   (JULY, 12 A.L.)

INSTRUCTIONS:

  SPECIFY THE BASE FOR THE FINAL, EVALUATED EXPRESSION.
  FOLLOW THAT WITH A SEMICOLON ';'.
  USE (, ), +, -, * AND / AS USUAL, BUT PRECEDE EACH
  NUMBER WITH ITS BASE CODE SYMBOL (DEFAULT=DECIMAL).  E.G.:

      H; 3 + (B110 * H-D)

  WILL YIELD THE HEX EQUIVALENT OF DECIMAL 3 PLUS THE
  PRODUCT OF BINARY 110 TIMES NEGATIVE HEX D.

  IF THERE IS NOTHING AFTER THE SEMICOLON, THE RESULTS OF
  THE PREVIOUS INPUT WILL BE CONVERTED TO THE DESIRED BASE.

  INTEGERS ONLY, PLEASE.

BASE CODE SYMBOLS:

B = BINARY   O OR Q = OCTAL   D = DECIMAL (DEFAULT)   H = HEX

>H;H7000/2
HEXADECIMAL: 3800

>H;   ( H 7000 / 2
HEXADECIMAL: 3800

>;
DECIMAL: 14336

>B;B111011+255
BINARY: 00000001 00111010

>H;
HEXADECIMAL: 013A

>B;HFFFF-4096)/2
BINARY: 01110111 11111111

>;
DECIMAL: 30719

>D;(HFFF-4096)/2
DECIMAL: - 1

>;(HFFFF-4096)/2
DECIMAL: 30719

>H
  ^
ERROR!  EXPECTING SEMICOLON

>2*((B1101+HAF)/16)
 ^
ERROR!  EXPECTING BASE CODE SYMBOL

>;2*((B1101+HAF)/16
DECIMAL: 22

>H;4096
HEXADECIMAL: 1000

>B;
BINARY: 00010000 00000000

>Q;
OCTAL: 010 000

>H;Q010000
HEXADECIMAL: 1000

>H;32049-(H7000/4
HEXADECIMAL: 6131

>B;
BINARY: 01100001 00110001

>D;5A-H20
    ^
ERROR!  INVALID CHARACTER FOR BASE 10

>;H5A-H20
DECIMAL: 58

>
```

Arithmetic, cont'd...

The program was written in Compucolor Disk Basic 8001 V6.78 on my Compucolor II and, as it stands, requires a bit over 9K RAM, not counting Basic itself. (This can be reduced considerably by deleting the REMarks and blank lines.) Multiple statements on a line are separated by colons. Because of round-off errors with the ABS and INT functions of Basics with floating point routines, this program occasionally converts a number into a string expression with the STR$ function and then converts it back to a numeric expression with the VAL function. (Lines 1540 and 1670.) If your Basic provides for integer-only arithmetic, then you needn't bother with such conversions.

The program should require only minor alterations for any other Microsoft Basic. Specifically, the various PLOT statements (except for line 20) are color commands, so if you are working in black and white you can omit them. □

```
READY

LIST

9 REM Set colors.
10 PLOT 6,6
18
19 REM Regular character height. Flag bit off.
   Scroll mode. Erase screen.
20 PLOT 15,29,27,11,12
29
30 PRINT "BASE ARITHMETIC    ";
39
40 PLOT 23:PRINT "BY DAVID B. SUITS   ";
41 REM             General Studies
42 REM             Rochester Institute of Technology
43 REM             Rochester, NY  14623
44
50 PLOT 19:PRINT "(JULY, 12 A.L.)"
59
60 REM
61 REM   NS( )...Number Stack.  Each time a number is found in
62 REM           input string, it is pushed onto this stack.
63 REM   NP......Number stack Pointer.  Points to most recent
64 REM           entry to Number Stack.
65 REM   OS$( )..Operator Stack.  Operators from input
66 REM           string are pushed onto this stack.
67 REM   OP......Operator stack Pointer.
68 REM   S( )....Strengths of operators, where the indices are
69 REM           determined by subtracting 40 from the ASCII
70 REM           values of the operators.  Thus:
71 REM
72 REM             OPERATOR    ASCII   S(i)    STRENGTH
73 REM                (          40      0        0
74 REM               xxx        xxx      1       xxx
75 REM                *          42      2        3
76 REM                +          43      3        1
77 REM               xxx        xxx      4       xxx
78 REM                -          45      5        2
79 REM               xxx        xxx      6       xxx
80 REM                /          47      7        4
81 REM
82 REM   BP......Buffer Pointer.  Points to position in input
83 REM           string of character being scanned.
84 REM   C$......Present Character being scanned.
85 REM   U$......Unary operator (+ or -).  If there is no
86 REM           unary operator to apply, then U$="!".
87 REM   E.......Error flag (see subroutine at 1400) in case
88 REM           a numerical expression contains an invalid
89 REM           character.
90 REM   B.......Base 2, 8, 10 or 16 of a number in the input
91 REM           expression.
92 REM   BASE....BASE 2, 8, 10 or 16 of the final, evaluated
93 REM           expression.
94 REM   L.......Length of input string.
95 REM
96
99 REM Clear some string space
100 CLEAR 100
108
109 REM Limit of 64 operators and 64 numbers.
    Surely that's sufficient!
110 DIM OS$(64),NS(64)
118
119 REM   Set strengths of operators
120 FOR J= 0TO 7:READ S(J):NEXT :DATA 0,0,3,1,0,2,0,4
127
128 REM   String of zeros for pretty-printing the output.
129 REM   Hexadecimal string used when changing bases.
130 Z$= "0000000":HX$= "0123456789ABCDEF"
137
138
139
140 PRINT :PRINT "INSTRUCTIONS:":PRINT
149
150 PRINT "  SPECIFY THE BASE FOR THE FINAL, EVALUATED EXPRESSION."
160 PRINT "  FOLLOW THAT WITH A SEMICOLON ';'."
170 PRINT "  USE (, ), +, -, * AND / AS USUAL, BUT PRECEDE EACH"
180 PRINT "  NUMBER WITH ITS BASE CODE SYMBOL
    (DEFAULT=DECIMAL).  E.G.:"
190 PRINT
200 PRINT "          H; 3 + (B110 * H-D)"
210 PRINT
220 PRINT "  WILL YIELD THE HEX EQUIVALENT OF DECIMAL 3 PLUS THE"
230 PRINT "  PRODUCT OF BINARY 110 TIMES NEGATIVE HEX D."
240 PRINT
250 PRINT "  IF THERE IS NOTHING AFTER THE SEMICOLON,
    THE RESULTS OF"
260 PRINT "  THE PREVIOUS INPUT WILL BE CONVERTED
    TO THE DESIRED BASE."
270 PRINT :PRINT "  INTEGERS ONLY, PLEASE."
279
280 PRINT :PLOT 18:PRINT "BASE CODE SYMBOLS:":PRINT
289
290 PRINT "B = BINARY    O OR Q = OCTAL
    D = DECIMAL (DEFAULT)    H = HEX"
297
298 REM

         Each input starts here.

299 REM   Zero the Buffer & stack Pointers. Set Error flag=0.
300 BP= 0:NP= 0:OP= 0:U$= "!":E= 0
309
310 PRINT :PLOT 21:PRINT ">";:PLOT 18:INPUT "";I$:L= LEN (I$)
320 GOSUB 1300:IF C$= ";"THEN BASE= 10:GOTO 420
330 IF C$= "H"THEN BASE= 16:GOTO 400
340 IF C$= "D"THEN BASE= 10:GOTO 400
350 IF C$= "O"OR C$= "Q"THEN BASE= 8:GOTO 400
360 IF C$= "B"THEN BASE= 2:GOTO 400
368
369 REM   No base code symbol found & not default, so Error.
370 GOTO 1700
397
398 REM   If end of input has been reached at this point,
399 REM   then there's an Error.
400 IF BP= LTHEN BP= BP+ 1:GOTO 1710
409
410 GOSUB 1300:IF C$< > ";"THEN 1710
417
418 REM   If end of input string has been reached after the ';'
419 REM   then give results of last evaluation but in new base
420 IF BP= LTHEN NP= 1:GOTO 800
427
428 REM   Now that we have the output base code symbol (or
429 REM   default to decimal), parse the rest of the expression.
430 IF BP= LTHEN 800:REM  All done!
438
439 REM   Check for Unary operator + or - for default=decimal.
440 GOSUB 1300:IF C$= "+"OR C$= "-"THEN U$= C$:B= 10:GOTO 550
448
449 REM   If no base code symbol, then assume decimal.
450 IF C$= > "0"AND C$< = "9"THEN B= 10:GOTO 550
458
459 REM   There must be either "(" or base code symbol.
460 IF C$= "B"THEN B= 2:GOTO 520
470 IF C$= "O"OR C$= "Q"THEN B= 8:GOTO 520
480 IF C$= "D"THEN B= 10:GOTO 520
490 IF C$= "H"THEN B= 16:GOTO 520
500 IF C$= "("THEN OP= OP+ 1:OS$(OP)= C$:GOTO 430:REM
    Push "(" onto stack.
508
509 REM   Error!
510 GOTO 1700
517
518 REM   If end of expression is reached then user did not
519 REM   enter a number.
520 IF BP= LTHEN BP= BP+ 1:GOTO 1720
528
529 REM   Check for Unary operator.
530 GOSUB 1300:IF C$= "+"OR C$= "-"THEN U$= C$:GOTO 520
538
```

```
539 REM   Check for invalid character.
54Ø IF C$< "Ø"OR (C$> "9"AND (C$< "A"OR C$> "F"))THEN 172Ø
548
549 REM
```

Get the number one character at a time & store as N$

```
55Ø N$= ""
56Ø N$= N$+ C$:IF BP= LTHEN GOSUB 14ØØ:GOTO 8ØØ
57Ø GOSUB 13ØØ:IF C$= > "Ø"AND (C$< = "9"OR (C$= > "A"AND
    C$< = "F"))THEN 56Ø
577
578 REM   Convert the number (now held as N$) into decimal and
579 REM   push it onto the Number Stack.
58Ø GOSUB 14ØØ:IF ETHEN 174Ø:REM   E<>Ø if there's an Error.
59Ø IF C$< > ")"THEN 64Ø
597
598 REM   ")" is scanned, so until Operator Stack is empty
599 REM   or has "(", apply last operator to top 2 numbers.
6ØØ IF OP> ØAND OS$(OP)< > "("THEN GOSUB 16ØØ:GOTO 6ØØ
6Ø8
6Ø9 REM   If top of Operator Stack has "(", then pop it off.
61Ø IF OS$(OP)= "("THEN OP= OP- 1
62Ø IF BP= LTHEN 8ØØ
63Ø GOSUB 13ØØ:GOTO 59Ø
637
638 REM   Now we're expecting an operator.
64Ø IF C$< > "/"AND C$< > "*"AND C$< > "+"AND C$< > "-"THEN 173Ø
65Ø IF OP= ØTHEN 69Ø
658
659 REM   Get strength of operator on top of Stack.
66Ø S1= S(ASC (OS$(OP))- 4Ø)
668
669 REM   Get strength of operator being scanned.
67Ø S2= S(ASC (C$)- 4Ø)
676
677 REM   If strength of S1 => strength of S2 then apply
678 REM   operator on Stack to top 2 numbers on Number Stack
679 REM   before pushing new operator onto Operator Stack.
68Ø IF S1= > S2THEN GOSUB 16ØØ:GOTO 65Ø
69Ø OP= OP+ 1:OS$(OP)= C$
698
699 REM   Now go back for another number.
7ØØ GOTO 43Ø
797
798 REM
```

Print out the final expression.

```
799 REM   First check for Errors.
8ØØ IF ETHEN 174Ø:REM   E<>Ø if there's an Error.
8Ø8
8Ø9 REM   Pop any "(" off Operator Stack.
81Ø IF OS$(OP)= "("THEN OP= OP- 1:GOTO 81Ø
82Ø IF NP= ØTHEN BP= BP+ 1:GOTO 172Ø
827
828 REM   If there's still an operator but only one number,
829 REM   then there's an Error.
83Ø IF NP= 1AND OP= 1THEN BP= BP+ 1:GOTO 172Ø
837
838 REM   While there are operators left, apply them in turn
839 REM   to the top 2 numbers on the Number Stack
84Ø IF OP> ØTHEN GOSUB 16ØØ:GOTO 8ØØ
847
848 REM   Get the absolute value of the number on the Number
849 REM   Stack. Use the STR$ function to avoid round-off errors.
85Ø N= VAL (STR$ (ABS (NS(1))))
858
859 REM   Now convert the number to desired output base.
86Ø R$= ""
87Ø A= INT (N/ BASE)
88Ø R= N- A* BASE:R$= MID$ (HX$,R+ 1,1)+ R$:IF A>
    ØTHEN N= A:GOTO 87Ø
9Ø8
9Ø9 REM   Format and print out the result.
91Ø PLOT 22:ON INT (BASE/ 5)+ 1GOSUB 96Ø,1ØØØ,1Ø4Ø,1Ø5Ø
92Ø IF NS(1)< ØTHEN R$= "- "+ R$
93Ø PLOT 19:PRINT R$
937
938 REM   If ABS(number) > 65535, it will not format
939 REM   correctly, so give overflow error.
94Ø IF ABS (NS(1))> 65535THEN 176Ø
95Ø GOTO 3ØØ
959
96Ø PRINT "BINARY: ";
97Ø IF LEN (R$)< 9THEN R$= RIGHT$ (Z$+ R$,8):GOTO 99Ø
98Ø R$= RIGHT$ (Z$+ R$,16):R$= LEFT$ (R$,8)+ " "+ RIGHT$ (R$,8)
99Ø RETURN
999
1ØØØ PRINT "OCTAL: ";
1Ø1Ø IF LEN (R$)< 4THEN R$= RIGHT$ (Z$+ R$,3):GOTO 1Ø3Ø
1Ø2Ø R$= RIGHT$ (Z$+ R$,6):R$= LEFT$ (R$,3)+ " "+ RIGHT$ (R$,3)
1Ø3Ø RETURN
1Ø39
1Ø4Ø PRINT "DECIMAL: ";:RETURN
1Ø49
1Ø5Ø PRINT "HEXADECIMAL: ";
1Ø6Ø R$= RIGHT$ (Z$+ R$,4):RETURN
1298
1299 REM
```

Subroutine to bump Buffer Pointer & get next Character.

```
13ØØ BP= BP+ 1:C$= MID$ (I$,BP,1):IF C$= " "THEN 13ØØ
131Ø RETURN
1398
1399 REM
```

Subroutine to convert number in input string to decimal.

```
14ØØ LN= LEN (N$):N= Ø
14Ø9
141Ø FOR J= 1TO LN
142Ø   ON INT (B/ 5)+ 1GOSUB 146Ø,148Ø,15ØØ,152Ø
143Ø NEXT :IF ETHEN RETURN :REM   E<>Ø if there's an error.
1438
1439 REM   Check for Unary operator
144Ø IF U$< > "!"THEN N= VAL (U$+ STR$ (N)):U$= "!"
1448
1449 REM   Push Number onto Number Stack
145Ø NP= NP+ 1:NS(NP)= N:RETURN
1457
1458 REM   Check for invalid characters
1459 REM   Binary
146Ø IF MID$ (N$,J,1)> "1"THEN E= J:J= LN:RETURN
147Ø GOTO 153Ø
1478
1479 REM   Octal
148Ø IF MID$ (N$,J,1)> "7"THEN E= J:J= LN:RETURN
149Ø GOTO 153Ø
1498
1499 REM   Decimal
15ØØ IF MID$ (N$,J,1)> "9"THEN E= J:J= LN:RETURN
151Ø GOTO 153Ø
1518
1519 REM   Hex
152Ø IF MID$ (N$,J,1)= > "A"THEN V= ASC (MID$
    (N$,J,1))- 55:GOTO 154Ø
1529
153Ø V= VAL (MID$ (N$,J,1))
154Ø N= VAL (STR$ (N+ V* B^ (LN- J))):RETURN
1597
1598 REM
```

Apply latest operator to top 2 numbers on Number Stack.

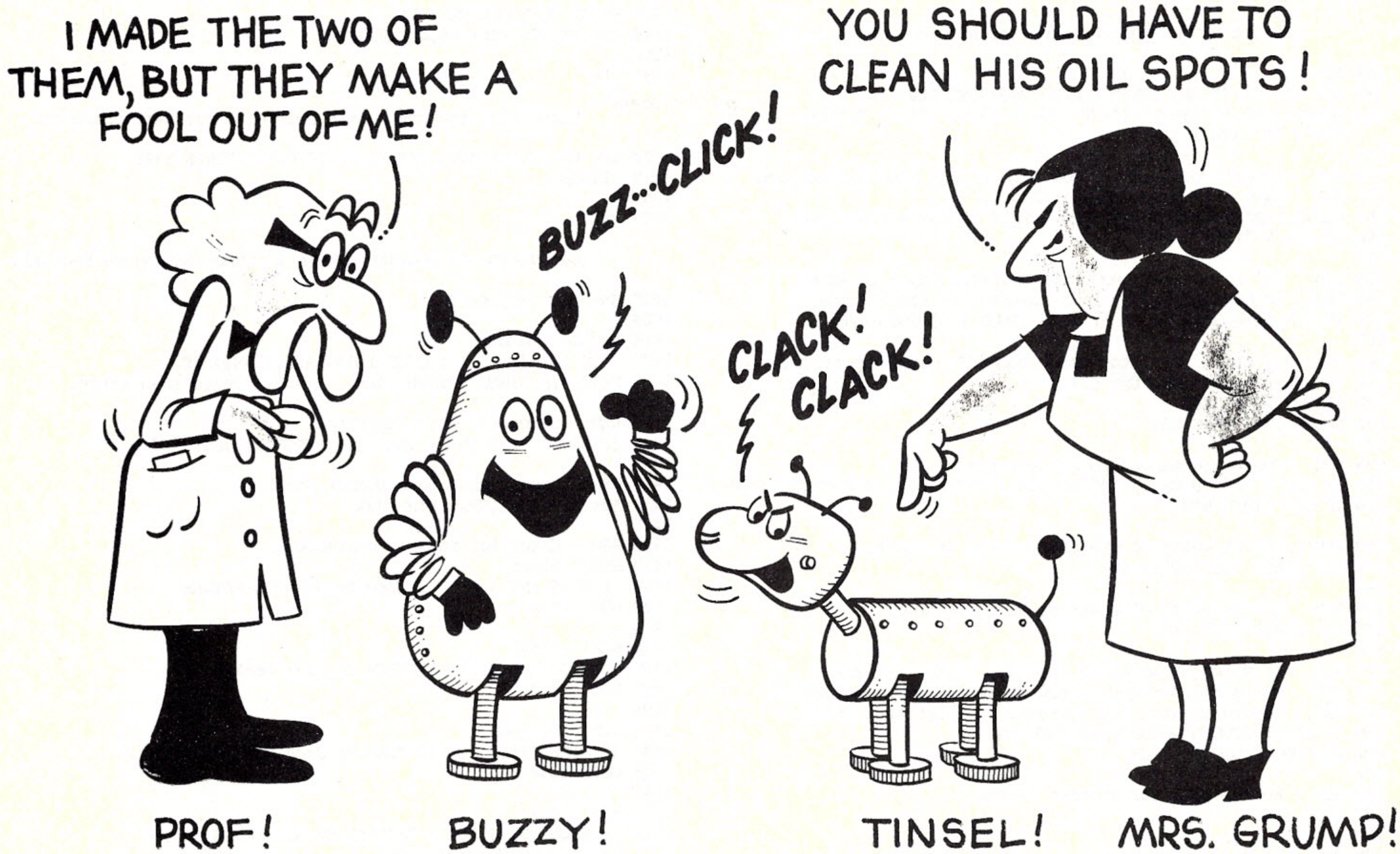
```
1599 REM   Get the 2 numbers.
16ØØ N1= NS(NP- 1):N2= NS(NP)
16Ø8
16Ø9 REM   Apply the operator.
161Ø O= ASC (OS$(OP))- 4Ø
162Ø ON OGOSUB 168Ø,164Ø,165Ø,168Ø,166Ø,168Ø,167Ø
1628
1629 REM   Pop operator & 2 numbers. Push new number onto stack.
163Ø OP= OP- 1:NP= NP- 1:NS(NP)= N:RETURN
1639
164Ø N= N1* N2:RETURN
165Ø N= N1+ N2:RETURN
166Ø N= N1- N2:RETURN
167Ø N= VAL (STR$ (INT (N1/ N2))):RETURN
1678
1679 REM Error message used during debugging
168Ø PLOT 17:PRINT "ERROR AT O="O:END
1698
1699 REM
```

Error messages

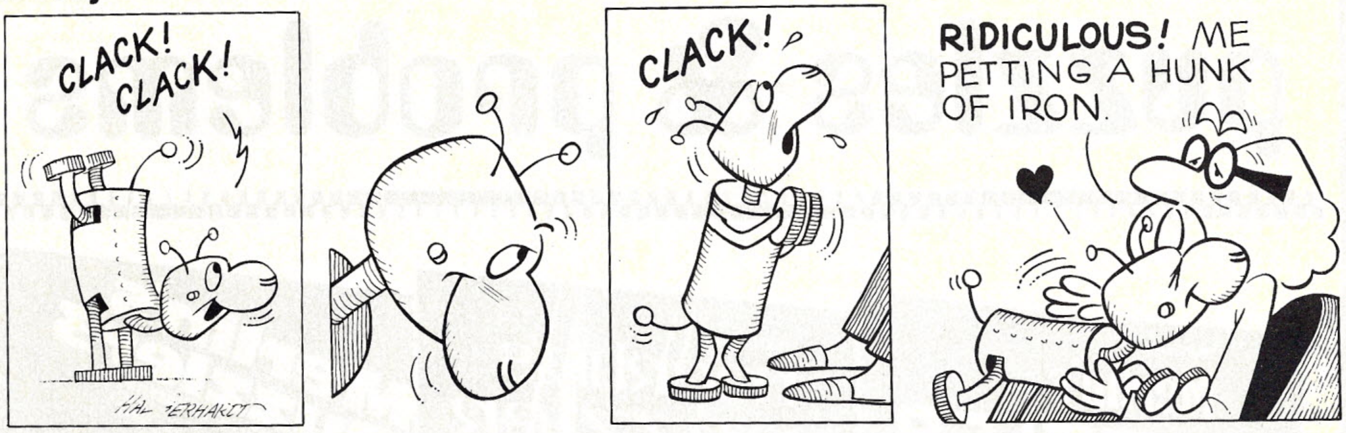
```
17ØØ GOSUB 177Ø:PRINT "BASE CODE SYMBOL":GOTO 3ØØ
17Ø9
171Ø GOSUB 177Ø:PRINT "SEMICOLON":GOTO 3ØØ
1719
172Ø GOSUB 177Ø:PRINT "NUMERICAL EXPRESSION":GOTO 3ØØ
1729
173Ø GOSUB 177Ø:PRINT "OPERATOR":GOTO 3ØØ
1739
174Ø PLOT 17:PRINT TAB( (BP)- (LN- E- (BP< L)))"^"
175Ø PRINT "ERROR!  INVALID CHARACTER FOR BASE"B:GOTO 3ØØ
1759
176Ø PLOT 17:PRINT :PRINT "OVERFLOW IN EVALUATED
    EXPRESSION":GOTO 3ØØ
1769
177Ø PLOT 17:PRINT TAB( BP)"^":PRINT "ERROR!  EXPECTING ";:RETURN
READY
```

BUZZY

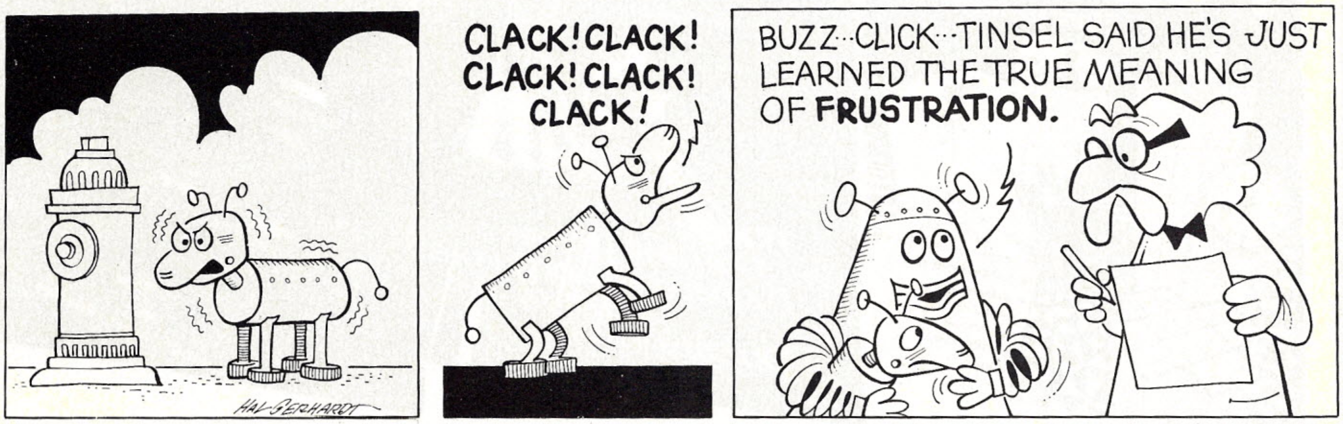
Hal Gerhardt



Buzzy Hal Gerhardt



Buzzy Hal Gerhardt



puzzles & problems

The year end holidays are rapidly approaching and Merlin thought that it would be a good idea to remind our readers that one of the best presents that they can give themselves, and their friends, would be a book or two from the famous Merlin's Puzzler series. The following review, covering the first two volumes of this series, appeared in GAMES Magazine.

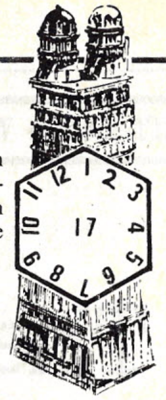
"Puzzle books are nothing new, and neither are the puzzles in them. But what sets *Merlin's Puzzlers* apart from the crowd is the style and imagination with which the material is presented. Townsend has adapted hundreds of the world's most intriguing puzzles, games, and magic tricks to original formats. In Volume 1 he calls upon Sherlock Holmes to pose the problems to Watson, and the Mad Hatter and Humpty Dumpty (among others) to confuse and confound "Alice in Puzzleland." Other fantasy trips lead to Merlin's (fictitious) library, and to Maskelyne and Cooke's Egyptian Hall (London's home of magic in the 1890s) for an anachronistic look at Houdini, Keller, Thurston, et al., in action."

"Merlin 1 and 2 are richly illustrated with old woodcuts, lithographs, prints, posters, and playbills from Townsend's collection. Thus, where Holmes appears, so do Sidney Paget's original illustrations from *The Stand,* and Alice is accompanied by John Tenniel's drawings from the works of Lewis Carroll."

"Many of the puzzles have not been heard from in years. Others are perennial favorites."

We thank GAMES Magazine for that generous review. As for Volume 3, well, all I can say is that it is just like 1 and 2, but, according to Merlin, better. All three of these books are large in size, 8½" x 11", and, each one contains 128 pages. All are obtainable from Creative Computing. Numbers 1 and 2 are available as a set for $7.50. Number 3 costs just $4.50. Order using handy postcard order form in the back of the magazine.

The Time Out Puzzle

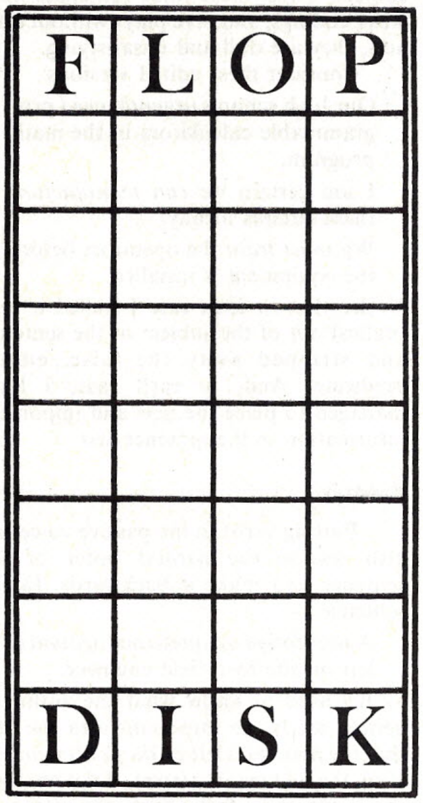


During The Festival Of Puzzles, held once a year on The Isle Of Merlin, they decorate the convention city with huge puzzles. In the picture shown here we see, hanging on the Park Plaza office building, a gigantic six-sided clock. "The puzzler may run out of time attempting to solve this puzzle," reads the convention program. "The challenge is simple. Merely rearrange the numbers on the clock so that the sum of each of the six sides totals 17." All right puzzlers, the clock is running! (From "Merlin's Puzzler 1")

Change-The-Word-Puzzles

Change-the-word puzzles seem to be among the most popular puzzles with our readers. Willard Wordsworth reports that the response from the "PLAY to WORK" challenge in July's "Puzzles and Problems" has provided us with several solutions to his challenge to the readers to find a solution to the problem in less than 8 moves. Some of the answers he received follow: From Ann Haase, of La Crescent, Minn. we have PLAY, PLAT, PEAT, PERT, PERK, PORK, WORK; From Robert E. McLintock, of Houston, Texas and Bryan Brwer, of Ashford, WA, we have PLAY, PLAT, PEAT, PEAK, PEAK, PERK, PORK WORK; David S. Alexander, of Wilmette, Ill., writes PLAY, PLAT, PEAT, PERT, PORT, PORK, WORK; Warren Spivak of Brooklyn, New York sends PLAY, CLAY, CLAP, CLOP, COOP, COOK, CORK, WORK; From Russell Olin, of Orlando, FL., we have PLAY, PLAT, PLOT, CLOT, COOT, COOK, CORK, WORK. Scott Mr. Jefferies, of Augusta, Ga., writes PLAY, PLOY, PLOT, BLOT, BOOT, BOOK, COOK, CORK, WORK. We thank you, one and all.

From Robert E. McLintock, who sent us one of the solutions reprinted above, we have a new change-the-word puzzle for Willard to add to his collection. Our readers are challenged to change the word FLOP to the work DISK in just 7 moves (or less). Here we go again! Willard will be looking forward to your answers. (For his efforts Mr. McClintock will receive a copy of "Merlin's Puzzler 1").



The Over-Polite Guests

Seven gentlemen met to dine at a restaurant, when a question arose as to precedence, no one desiring to take what were regarded as the more honourable seats. To settle the matter, one of them proposed that they should dine together every day until they had respectively occupied all possible positions at the table; and the suggestion was accepted.

How often must they dine together to answer the above conditions?

(This Puzzle is from "Merlin's Puzzler 2").

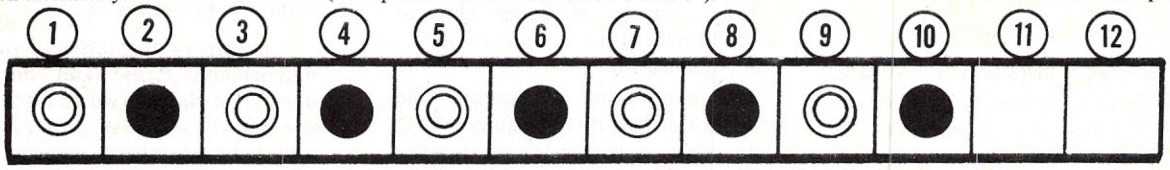
Rule on cardboard a rectangular figure divided into twelve squares, as shown below, and in the first ten spaces, beginning from the left hand, dispose ten counters, red and black alternately.

The puzzle is to move *two adjacent counters at a time* so that the five red and the five black counters are grouped, each color together without any interval, and this must be done *in five moves only.* At the close of the operation the ten counters should be grouped five black and five red.

They are then to be worked back again, after the same fashion, to their original positions.

(This puzzle is from "Merlin's Puzzler 3").

Answers on page 224.



So ends another session of Puzzles and Problems. If you have a favorite puzzle you would like to share with the other readers of *Creative Computing* send it along to Merlin. If he uses it he will send you a copy of one of his books.

Until next time, good puzzling!

Your editor

Charles Barry Townsend

CHARLES BARRY TOWNSEND

Effective Writing

Edmond H. Weiss, Ph.D.

The Secret of the Sentence

Edmond H. Weiss, a communications consultant, teaches effective writing seminars for business, industry, and government. To contact him, call 609-795-5580.

Quick. Tell me where you should put the most important words in a sentence. If you answered, "the beginning," you're wrong.

One of the best kept secrets about effective writing — a secret your English teachers never told you — is that *a well-made sentence almost always saves its new and interesting information for THE END.* As Strunk & White put it in *The Elements of Style* —

> The proper place in the sentence for the word or group of words that the writer desires to make most prominent is usually the end.

The fact that so few computer people and other amateur writers realize this accounts for much of the tediousness in writing about computers and technology. A good many of the sentences in print, even though they are grammatically correct, have all their important information in their first few words. They do not so much end as fall off and die.

And, to make matters worse, during a first draft it is *natural* for writers — even the very best ones — to blurt out their main information at the beginnings of their sentences. Those writers who never revise or edit (and you know who you are) send out a lot of backwards sentences.

Let me illustrate.

Inverted Sentences

This sentence is just plain backwards:

> Improved morale is the main effect of the word processing system mentioned above.

Instead, it should read:

> The main effect of the word processing system mentioned above is (Feel the suspense?) improved morale.

Here's another backwards item:

> Documentation techniques is a subject lacking in the computer sciences curriculum.

Instead, it should read:

> A subject lacking in the computer sciences curriculum is documentation techniques.

In nearly every case these inverted sentences are the result of our making sure that we get the main point down, right away, before we forget what we wanted to write. (In a first draft, therefore, they are excusable.) But they need to be revised, so that they build in interest and make the reader want to find out what happens next.

Empty Predicates

The end of the sentence, the place where the new and emphatic information should usually appear, is generally the *predicate.* (That is, a verb, followed by some modifiers, or an object, or some complement to the subject.)

The "action" in a sentence should be in the predicate, preferably in the verb. Unfortunately, though, the same writers who blurt out the main information first also leave no important information to put in the predicate.

Consider these sentences in which the authors spoke their entire piece before they even hit the verb:

> The urgent need for programmable calculators in the high school math program . . . exists.
>
> The certainty of being able to manufacture these circuits locally... is emphasized.
>
> The importance of training the operators before the equipment is installed . . . is called to your attention.

Although each of these sentences is grammatically well-formed, each also contains an *empty predicate,* a predicate in which nothing interesting happens because the author has said everything in the subject. Thus, these three sentences have weak endings, and, like plays without final acts, they are dull and unsatisfying.

Consider these edited versions:

> Our high schools *urgently need* programmable calculators in the math program.
>
> I am certain we *can manufacture* these circuits locally.
>
> We *must train* the operators before the equipment is installed.

Notice that in each case I pulled a verb (italics) out of the subject of the sentence and stripped away the false, empty predicate. And, in each case, I have managed to place the new and important information in the sentence *last.*

Passives

Putting verbs in the passive voice can also reverse the natural order of the sentence and make it backwards. In the sentence —

> A preventive maintenance manual is left on site by a field engineer.

— it's hard to know what the author is getting at. Is the important idea the fact that the manual is left by *the field engineer?* If so, the sentence is correct. But if *on site* is the important fact, then the sentence should read:

> A field engineer leaves a preventive maintenance manual on site.

And, if the emphasis is on the *manual* itself, then —

> A field engineer leaves on site a preventive maintenance manual.

Complex Sentences

Complex sentences contain one independent clause and one or more dependent clauses (which often begin with the words *because, since, when, as, although* and others). Because, in a well-written complex sentence, the main assertion is usually in the *independent* clause, the independent clause should usually come last.

For some reason though — perhaps an editor's reluctance to start with

Edmond H. Weiss, Ph.D., 1612 Crown Point Lane, Cherry Hill, NJ 08003.

anything other than the main subject or a writer's neurotic fear of beginning a sentence with "because" — many of the complex sentences in the computer literature are backwards. For example:

> *Backwards*
> We have switched to a DBMS because we were spending too much for programming.
>
> *Correct*
> Because we were spending too much for programming, we switched to a DBMS.
>
> *Backwards*
> The customer will never accept this report even though the analyst likes it.
>
> *Correct*
> Even though the analyst likes this report, the customer will never accept it.

Of course, you could object that in some cases the author *chose* to emphasize the information in the dependent clauses, and that the "backwards" versions are better. If that were so, you would be right.

But, usually, that is not so. The way our sentences squirt out in the first draft is the way they stay, as though we had no power to influence them.

You Have the Power

Here is the main point: You, the writer, decide what you want to emphasize and then, while you revise, you contrive to have it appear at the end of the sentence. Almost no one can make these calculations during the first draft; we all must do them while we edit.

You have the power. Even if the first draft version has a will of its own, you are never a slave to your own first versions.

As it turns out, an easy way to decide what goes at the end of the sentence is to see what's talked about in the *next* sentence. The most fluent arrangement of thoughts is to have what is new in Sentence 1 (therefore at the end) become what is old or established in Sentence 2 (therefore at the beginning). (This sequence eliminates much of a problem that many of my clients complain about: "choppiness".)

So, the secret of sentences is to have them move from old to new, from familiar to unfamiliar. The proper arrangement depends on what has already been discussed in the previous sentences and what will be discussed next.

For fun, think of a case in which each of these versions would be correct:

A. Anticipating delays is the main function of a project management system.

B. The main function of a project management system is anticipating delays.

Next time: Too many words. □

The comments and opinions of the author are given for educational purposes only and are not meant to be legal advice. Specific legal questions should be referred to your personal attorney.

Harold L. Novick

As promised in last month's Forum here is the "rebuttal" by Mr. P. V. Piescik of Cuddly Software, Wethersfield, CT. Although the "rebuttal" contains some minor errors, and although there is disagreement with some of its statements, it is set forth as submitted to give a very interesting perspective.

April 20, 1980

Dear Mr. Novick,

Enclosed, at long last, is my "rebuttal" to the CompuChess decision.

Actually, I changed my approach, and this is more appropriately termed a "perspective," since I chose not to challenge the precedent, but rather accept it as the *status quo,* and point out (I hope) where it leaves software, and where it should be.

In the course of writing this, I consulted with another software design engineer, whose thoughts closely parallel mine, and contributed somewhat to my perspective. His work, however, is classified, beyond the general mention that it deals with airborne data systems, and is ostensibly unaffected by this whole subject. At least I can say that a minimum of two software-types (he and I) agree with my evaluation of the situation!

While my primary intention in writing this is to contribute (hopefully) to the creation of realistic protection of software properties, I leave publication to your discretion.

Sincerely,
P. V. Piescik

The CompuChess case (Data Cash Systems, Inc. v. J S and A Group, Inc., 203 USPQ 735 (N.D. Ill. 1979) proves that all software is totally unprotected under present copyright law! Regardless of any of the variety of individual opinions, pro or con, precedent has been established, with some horrible implications.

Several points were brought out in the decision, which show that the nature of software is not understood. It stands to reason that protection cannot be afforded to an entity which is best recognized by one's intuitive sense; intuition is a poor defense in Court. Copyright law, which is usually the means of attempted protection of software, explicitly excludes object code, which ultimately is the only executable, i.e., useful, i.e., necessary form of a program. A pirate need only translate protected (?) source programs from high-level or assembly language to machine language (object code) to circumvent even nominal protection. And an object program is not recognized as a copy of the source program from which it is produced by translation.

Incidental misunderstandings of computers, programmers, and software occur in the decision, they are moot in light of the lack of protection. However, if realistic coverage is to be provided, they must be resolved, and so will be discussed.

Assembly-language was equated by the Judge to machine-language. Machine language is obviously "understood" by the machine (computer), which understands only what it can execute. Machine-language is equivalent to executable object code; assembly-language source code is only input data to the assembler, for translation into object code.

The Judge also stated that assembly programs and object programs are difficult and impossible (respectively) to be understood by programmers, regardless of their training and/or experience. This statement would be laughable, were it not for the grave nature of its implications. Both are products of man's intellect; it is illogical to believe that the same intellect cannot comprehend them. Their only shortcoming is a lack of convenience, as their form is not ideal to facilitate human comprehension. While object code must retain its form to be interpreted by a machine, a great deal of latitude is provided by most assembly languages in the use of symbolic labels; it is possible and desirable to utilize these symbols and produce programs which are almost English, if a bit cryptic and stilted. These lowest-level forms of code are not incomprehensible, merely inconvenient. The computer can also be used to disassemble object code at least to the assembly level.

In the decision, object code in ROM is also classified as hardware, not software, and as such is a part of the computer circuitry. The same might be said for programs in RAM, since any object program is merely an ordered set of state settings for a multitude of binary switches both in the memory and in the processor and other components during execution. The progression of this line of thought may lead to the conclusion that there is no software, only hardware! ROM is simply another medium, only non-volatile.

If, however, the ROM is not seen as a medium for software, and therefore the object code is a hardware entity, the question of patentability arises. The patent is ostensibly the appropriate form of protection for hardware rights. Acquiring a patent is a rather complicated process, time-consuming, and much more expensive than registering a copyright. Among the qualifications for patentability are application and novelty, both of which may be in the range of difficult to impossible to demonstrate. Few programs are sufficiently new and different from previously written software, especially in an industry where compatibility and portability are very desirable. The application requirement may exclude all software of the systems class, as operating systems, I/O modules, language processors and various utilities are the tools of computer science. Tools of science do not qualify for patents. Patents might be more easily obtained to protect the "concept" of the program, which should be construed as the

process described by the program for performance (execution) of the process by the computer. The logical extension of this protection would prevent registration (i.e., patenting) of all new Basic interpreters, for instance, once the first interpreters were registered! Substitute the name of virtually any other program (Star Trek games, assembler, accounts receivable ledger, etc.) for "Basic interpreter," and this method is obviously too restrictive.

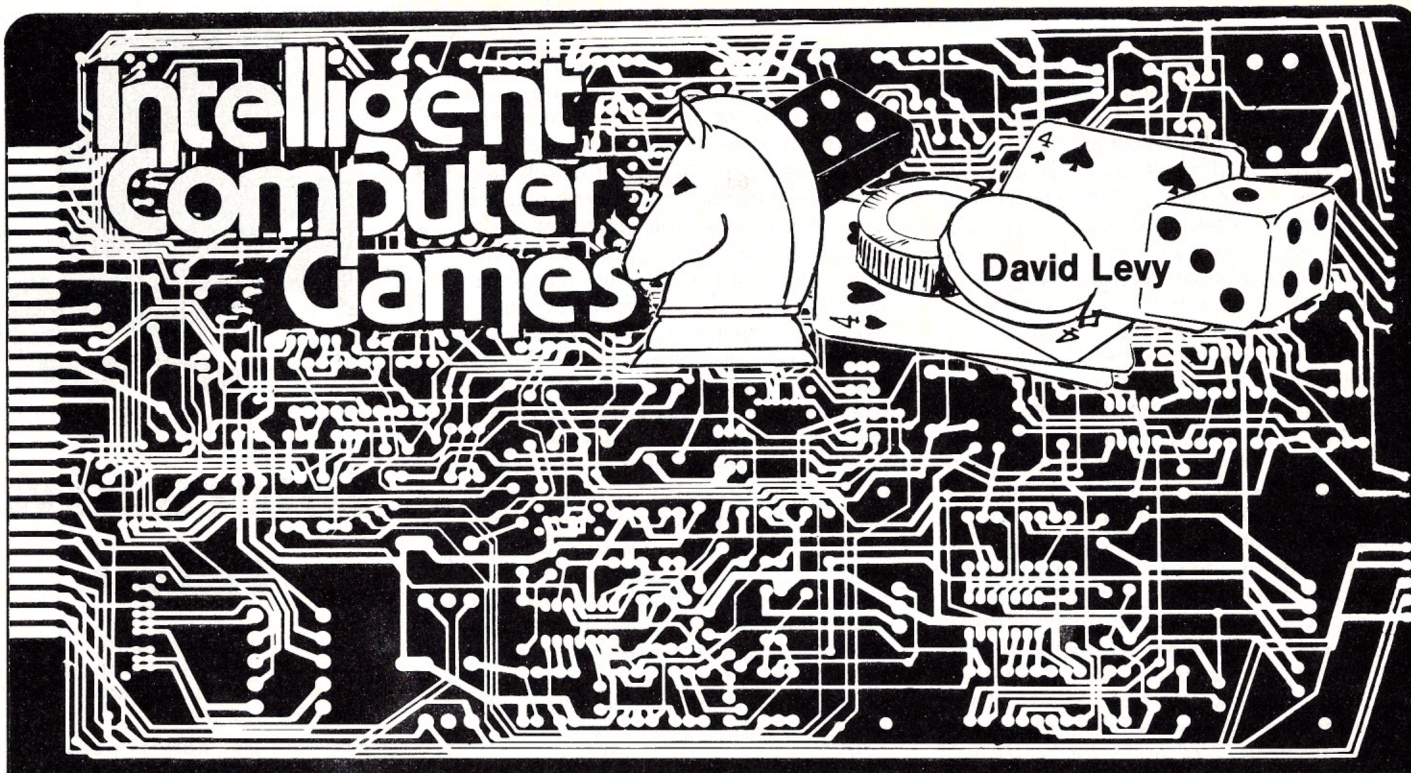
Computer and software are a radical departure from previous forms of machinery and communications, when taken as a whole. Even among computer scientists, many aspects of computing are extremely difficult to define, explain, and comprehend in deterministic terms. It should not be surprising, then, that attempts to force-fit software into the requirements of either copyright or patent protection fail amid controversy! Software is a communication of ideas between human and machine, spanning a tremendous difference in intelligence. To a machine which is no more than an amazing conglomeration of toggle switches (or their electronic equivalent) and which understands only 'on' and 'off', the alphabet, let alone the vocabulary, of man has the complexity of a five-dimensional universe. (Fortunately, the machine fairly adequately compensates for its lack of intelligence with its speed.) Object programs are man's ideas distilled to machine level; other languages (levels) are simply arbitrary limitations imposed upon man's vocabulary to facilitate mechanical translation, while allowing easier human recognition of the format of expression used to communicate the ideas. It is necessary, therefore, that several forms of a program exist simultaneously on a variety of media. The alternative is to require the destruction of all but one form, existing in a single medium, at any given time for any given program. This is counter-productive. It would require, for instance, that a tape copy of a program be erased when the program is loaded into memory, and re-created upon termination of the execution run; modifications requiring human intervention would require translation back to high-level code, and subsequent translation to object code. This would be extremely inconvenient, wasteful, and impossible to protect; several forms of protection would be required, which each in effect or dormant as the corresponding program form exists.

Protection which does not cover any and all forms and media of a program is useless. As previously mentioned, one need only to translate a protected form of a program to an unprotected form. Some copying is also necessary to protect the interests of the consumer, such as creating a copy in memory from other media to run the program, and back-up copies of the other media to prevent losses due to catastrophe. The need for this copying is recognized by the producer as acceptable. So, when is copying not acceptable?

Generally, profit and/or distribution are the deciding factors in determining the acceptability of copying. However, these are usually examined from the wrong standpoint: whether distribution is potentially profitable for the copier. Especially regarding computers, with the aboundance of clubs and user groups, this is the wrong criterion. Proliferation of copies among members may not even involve pecuniary transactions; each recipient simply provides his own tape or disk to receive a copy, and someone usually has access to an office machine to produce free copies of printed documentation. With software, the criterion should be "copying to avoid purchase (from the producer)" with any transaction between two or more parties viewed as being conducted with the intent to avoid purchase. This is where the producer loses in his attempt to recover his investment, which could lead to higher prices and less available software.

As parts of existing protective legislation will undoubtedly be the basis for extension to realistic coverage for software, yet another point deserves mention. Object code lacks "eye appeal," one of the qualifications required for copyright. However, in the CompuChess case, and perhaps copyright law itself, the computer is not recognized as a form of mechanical assistance, which is allowable as in the instance of microfilm/fiche media. Granted the computer is more complex than a microfilm viewer, but it is a machine assisting man's visual perception ("eye appeal") of the software, and should be recognized.

The need, then, is for a form of protection which recognizes a program as a polymorphic description of a process, ultimately to be executed by a machine. All forms should be recognized as equivalent by means of translation according to the provisions of the programming languages used, which is simple enough to demonstrate. Copying any or all forms of a program must be allowable in the interest of preventing catastrophic losses to the consumer; copying for distribution to other parties, with the intent to avoid purchase, regardless of profit or the lack of same, must be forbidden. Fixation of any or all forms in a particular medium should not be required, and in the case of media which are machine-readable only, the computer must be recognized as a form of mechanical assistance if visual perception of the work is a requirement. Algorithmic, and possibly, flowchart representations of the process, and the process *per se,* should not be restricted, as such restriction would serve to hamper scientific communication and progress. □



Correspondence is welcome. Letters with interesting questions and ideas will be used in the column along with a response. No personal replies can be made. Send to: David Levy, 104 Hamilton Terrace, London NW8 9UP, England

CHESS

Of all the games that have attracted the attention of computer programmers, chess must surely rank at the top of the list. This is partly because chess is considered by many to be *the* intellectual game, par excellence, and therefore the creation of a strong chess program can be equated with the creation of an artificial intellect. Another reason is that writing a chess program is a great challenge.

A measure of the popularity of computer chess programming may be judged from the history of computer chess contests. In 1970, in New York, there was a tournament in which all six of the contestants were computer programs. The event proved so popular that it has been repeated each subsequent year, at the annual conference of the Association for Computing Machinery, and by the end of the 1970s there were usually between 12 and 20 applications for places in the tournament. Now that personal computers are available in large numbers, chess contests are springing up specifically for small computers. Personal Computer World is organizing its third such tournament in London on September 4th-6th this year, and this event has the status of World Championship, and is being held under the auspices of the International Chess Federation — an indication that "real" chess players are now beginning to take computers seriously. Other small computer chess tournaments are being held in the USA, France and Germany. Probably there will be a dozen such events in the calendar by the end of 1981.

Because of the enormous interest in computer chess, a lot has been written on the subject. I have decided that this article and the following one will provide a history of the most important milestones in the field, and I shall discuss how the ideas employed in mainframe programs may be applied to computers. In a third article I shall discuss the current state-of-the-art of computer chess programming, with many examples taken from actual games.

In the Beginning

On March 9th, 1949, the American mathematician Claude Shannon delivered a paper at a New York conference. The paper was called *Programming a Computer for Playing Chess,* and it is remarkable that many of Shannon's original ideas have permeated through to the programs of today. He pointed out that there are some 10^{120} possible games of 40 moves (the average length of a master game), and that analyzing to this depth at the rate of one game per microsecond would take a computer 10^{90} years to make its first move! A similar, though even more emphatic argument, is that the number of possible chess games so far exceeds the number of atoms in the universe, that even were each atom to be replaced by a Cray 1 computer, it would take the whole system a rather long time to make the first move in a perfect game of chess.

Having dispensed with the notion of perfect play through exhaustive search, Shannon set about defining an evaluation function which would give a reasonably reliable estimate of which side held the advantage in a position, and by how much. His example of a crude evaluation function was:

$$200 \times (K_w - K_b) + 9 \times (Q_w - Q_b) + 5 \times (R_w - R_b) + 3 \times (B_w - B_b + N_w - N_b) + (P_w - P_b) - 0.5 \times (D_w - D_b + S_w - S_b + I_w - I_b) + 0.1 \times (M_w - M_b)$$

where K, Q, R, B, N and P represent the number of pieces of each type (king, queen, rook, bishop, knight and pawn), and the subscripts w and b refer to white and black. D is the number of doubled pawns (pawns of the same color on the same file); S is the number of backward pawns (pawns that cannot be defended by a pawn); I is the number of isolated pawns (pawns with no neighbor pawns of the same color); M is the measure of mobility (the number of legal moves at a player's disposal).

The king is given an arbitrary high value because loss of the king means loss of the game. The values of 9, 5, 3, 3 and 1 for the other pieces are the rule-of-thumb values which chess players learn early in their careers, though bishops are usually regarded as being more valuable than knights so in your chess program you might experiment with values of 3-1/4, 3-1/3 or even 3-1/2 for a bishop.

Games, cont'd. . .

Shannon's evaluation function is sufficient to provide a reasonable level of performance in a computer chess program. Chess, however, is a complex game, and Shannon recognized the need for the use of many other features if the evaluation function were to result in a strong program, and he suggested the following additional features:

1. Relative control of the center by pawns (white pawns at c4, d4, e4 and f4; black pawns at c5, d5, e5 and f5).

2. Weakness of pawns near your own king (e.g., advanced pawns in front of the king after castling).

3. Placing pawns on opposite colored squares from your own bishops to allow the bishops greater freedom of movement.

4. Passed pawns (i.e., pawns which have no enemy pawns in front of them, either on the same file or on adjacent files). These pawns can often become queens in the endgame.

5. Advanced knights (white knight at c5, d5, e5, f5, c6, d6, e6 and f6; black knights at c4, etc.), especially if protected by a pawn and free from attack.

6. Rooks on open or semi-open files (an open file is one with no pawns; a semi-open file has one pawn belonging to the opponent).

7. Rooks on the seventh rank. (A white rook on a7, b7, . . . , or a black rook on a2, b2, . . . , etc., can wreak havoc in the endgame by picking off the opponent's pawns.)

8. Doubled rooks (two rooks of the same color on the same file).

9. Pieces which are required for guarding functions and, therefore, committed and with limited mobility.

10. Attacks on pieces which give one player the option of exchanging.

11. Attacks on squares adjacent to the enemy king.

12. Pins. (A pin is a setup in which one piece may not move because of the loss of a piece which it is shielding. E.g., white bishop on g5, black knight on f6, black queen on e7. Black may not move his knight because of the loss of his queen — the knight is said to be pinned by the bishop.)

The addition of these features would provide a rather sophisticated evaluation function for middle game play, though as Shannon himself pointed out, different factors apply in the opening and (to a lesser extent) in the endgame.

When and How to Use The Evaluation Function

Shannon understood that it is only safe to use this type of evaluation function in positions which are relatively quiescent. If White makes a move capturing black's queen, it is not sensible to evaluate the resulting position without looking to see if black might be able to recapture white's queen in reply, or whether he might be able to checkmate. In fact it is meaningless to evaluate a position during a series of exchanges, unless the evaluation mechanism allows for the fact that further meaningful exchanges are possible. Chess players recognize quiescent positions intuitively, but computer programs have more difficulty because they cannot immediately determine which capturing moves and sequences are "obviously" wrong, in the way that a human chess master can.

Shannon called a fixed depth search strategy a "type A" strategy. He realized that in chess this type of strategy would lead to weak play, partly because of evaluating many non-quiescent positions once the fixed search depth had been reached, and partly because of the time required for exhaustive search (the alpha-beta algorithm had yet to be invented in 1949). Again he alluded to the thought processes of chess masters, and in particular to the work of the Dutch psychologist De Groot who recorded the spoken thoughts of chess masters as they analyzed a number of typical chess positions. Shannon concluded that in order to improve the speed and strength of the program it would be necessary to:

1. Examine forceful variations as far as possible and evaluate only in quiescent or quasi-quiescent situations.

2. Select the variations to be examined by some process so that the program does not waste a lot of time in totally fruitless variations.

Shannon called this type of strategy a "type B" strategy, and it is the Shannon B strategy which is used in almost all of the most successful programs of today. The key to the Shannon B strategy is the ability to determine which moves and variations are worth considering, a problem on which much has been written but rather little accomplished during the past three decades. When I examine a chess position I can usually make a reasonably good move after looking at only 50-100 nodes of the game tree. In order to play at the same level, the current world computer champion must examine over one million nodes. If it had the same ability to discern which variations are important, it would be able to defeat Bobby Fischer.

In order to help decide whether a move is worth exploring, Shannon suggested the use of a function which would return a large value for forcing moves (captures, checks and attacking moves), medium values for defensive moves, and low values for all other moves. As the depth of search increased, the requirements of this function would be set higher so that fewer and fewer subvariations would be examined. This approach has proved successful in a number of strong chess programs, and can easily be implemented on a micro. One simple method of doing so would be to examine all moves at the root of the tree, then only the most important 90% of moves at ply-1 ("importance" being determined by Shannon's discrimination function), then only the most important 70% at ply-2, 50% at ply-3, 30% at ply-4, and 10% at ply-5 and beyond, down to the limits imposed by search time restrictions or to quiescent positions. My method would call for an examination of fewer than 10% of the number of nodes normally examined in a 5-ply search, and the 90% saving could be used either to increase the sophistication of the evaluation function (which would also make it slower), or to increase the maximum depth of search in tactical situations.

Another idea suggested by Shannon was the use of typical chess positions or fragments of positions, for which a particular move or moves is known to be effective. Chess masters use this type of information all the time. They recognize a situation and immediately start to examine a move which they know has often proved strong in similar positions. Of course it will not always be the case that exactly the same move is best in a slightly different situation, but as we have seen in previous articles it is extremely important to examine the most likely moves early in the search process. Unfortunately, the only substantial example of this approach was a dismal failure. A strong American chess master, Charles Kalme, implemented a method involving "snapshots" of chess situations. His work was discussed in a *Scientific American* article in 1973, but shortly thereafter his program fared dismally in the annual ACM computer chess tournament in Atlanta, and little has been heard of the program since then. Perhaps this is one example of a technique used by humans which will be difficult to employ in a computer program. In any event, Kalme's failure should not worry the small computer user, since the amount of memory required to use the snapshot approach would be prohibitive at today's prices.

The Bernstein Program

Shannon's work was purely theoretical in nature. He did not write a chess program to test his ideas, though if he had I suspect that his program would have been stronger than some commercially available programs which are now on the market. The first example of a program playing full games of chess was seen in the late 1950s. This program was written for the IBM 704 computer by Alex Bernstein of IBM, and three colleagues. Since your own machines will all be considerably more powerful than an IBM 704, any of you who write chess programs ought to be able to do at least as well.

Q. Your students are gathering around the several PET computers in your classroom. And they all are hungry for hands-on turns at the keyboards. Some students are just beginning to understand computers; others are so advanced they can help you clean up the programs at the end of the period. How do you set up a job queue, how do you keep the beginners from crashing a program, how do you let the advanced students have full access? And how do you preserve your sanity while all this is going on?

A. With the Regent.

Q. What is the Regent?

A. The ultimate in classroom multiple PET systems. A surprisingly inexpensive, simple, effective way to have students at all levels of computer capability work and learn on a system with up to 15 PETs while the instructor has complete control and receives individual progress reports.

Q. SUB-it? Proctor? What are they?

A. **The SUB-it** is a single ROM chip (on an interface board in the case of the original 2001-8 models) that allows up to 15 PETs to be connected to a common disk via the standard PET-IEEE cables. The Commodore 2040, 2050 or 8050 dual disks and a printer may be used.

The SUB-it prevents inadvertant disruption when one unit in a system is loading and another is being used.

The Proctor takes charge of the bus and resolves multiple user conflicts. Each student can load down from the same disk but cannot inadvertently load to or wipe out the disk. Good for computer aided instruction

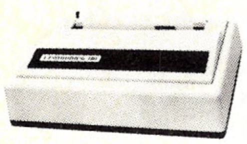
Q. How expensive are these miracles?

A. We think the word is **inexpensive**. The **Regent** system is **$250** for the first PET; **$150** for each additional PET in the system. The **SUB-it** is **$40**. (Add an interface board at **$22.50** if the PET is an original 2001-8.) And the **Proctor** is $95.

Phone or write for information. We'll be delighted to answer any questions and to send you the complete information package.

Skyles Electric Works
231 E South Whisman Road, Mountain View, California 94041 (415) 965-1735

CIRCLE 254 ON READER SERVICE CARD



PRINTER PRICES DROPPED!

Model 730 .... **$649** (reg. $795)
- Parallel Interface
- Upper & Lower Case
- 100 cps — 80 character line
- 5 x 7 Free Flight Ballistic Head
- Roll, Single-Sheet, & Tractor Feed Paper

Model 737 .... **$849** (reg. $995)
- Proportional Spacing
- Right Justification
- Underline & Expanded Print
- Bidirectional Paper Motion (for superscript and subscript)
- High Density Dot Matrix (N x 9 free flight print head) (18 possible horizontal dot places)
- Roll, Single-Sheet, & Tractor Feed Paper

Computerware Offers:
- Full Factory Warranty
- Immediate Delivery
- Dependability - as a stocking Centronics distributor for 2 yrs
- The latest models at affordable prices

TO ORDER
Phone orders are invited. Use VISA, Master Charge, or send cashiers check or money order drawn on a U.S. bank. Add shipping and handling (2% for 730, 4% for 704) or printer will be sent freight C.O.D.

COMPUTERWARE
1512 Encinitas Blvd., Box 668
Encinitas, CA 92024
(714) 436-3512

CIRCLE 125 ON READER SERVICE CARD

GREEN PLUS®

only **$14.95**

ADD A NEW DIMENSION TO YOUR VIDEO DISPLAY.

Transform your black and white monitor (or TV) into a **GREEN VIDEO DISPLAY** with **GREEN + PLUS,** a high quality custom cut transparent shield which attaches in just minutes to your monitor. No tools are required.
• specially designed to reduce eye fatigue • a sharper, crisper image • designed by an Industrial Designer and a Software Programmer • provides maximum clarity

SPECIAL INTRODUCTORY PRICE

Please send me ________ **GREEN + PLUS** screens at $14.95 each. I have included a payment total of $____________.
(Mass. residents must add 5% sales tax).
I am paying by
☐ Master Charge ☐ Visa ☐ Money Order ☐ Check.
Charge Card # ____________________
Expiration Date ______/______
Cardholder's Signature ____________________
Name ____________________
Address ____________________
City ______________ State______Zip______
Screen size.... WIDTH:__________ DEPTH:__________
Monitor (or TV) Name ____________________

CRT Computer Research & Technology
P.O. BOX 783
ISLINGTON STATION
WESTWOOD, MASS. 02090

CIRCLE 147 ON READER SERVICE CARD

Games, cont'd. . .

Bernstein et al, employed four features in their evaluation function: Mobility, area control, king defense and material. Area control was defined as the number of squares controlled completely by each side, while king defense counted the number of controlled squares around the king. Their material feature was weighted with the ratio of its own material to that of its opponent, in order to encourage the program to exchange material when ahead and to discourage it from exchanging when behind. This simple heuristic is extremely well known, but not all programmers consider it worthwhile to implement it, and possibly because programs play worse in the endgame than they do during the middle-game!

Moves were generated in response to a number of questions:

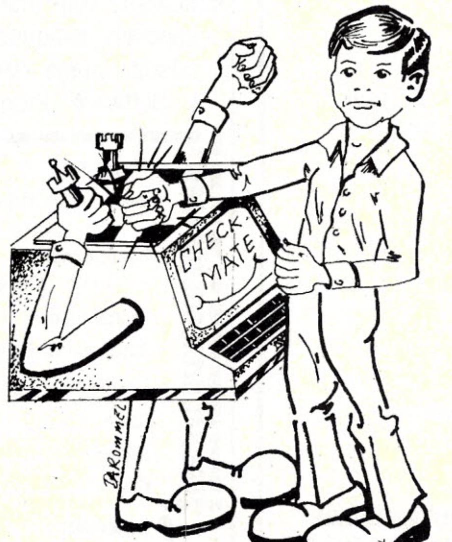
1. Is the king in check?
2. Can material be gained, lost or exchanged (i.e., can the program make an equal or advantageous capture, or is it threatened with material loss)?
3. Is castling possible?
4. Can a minor piece be developed?
5. Can key squares be occupied? (Key squares are those squares controlled by pawns.)
6. Can an open file be occupied?
7. Can any pawns be moved?
8. Can any piece be moved?

If the answer to question 1 is "yes," the program generates moves that reply to the check, and these moves are put into a "plausible move table." If the answer to question 1 is "no" the program goes to question 2, and so on. If the answer to question 3 is "yes," no other moves beyond question 3 are examined, as castling was considered so important that no other moves except replies to check and material changing moves are of greater importance to the program.

At the beginning of the game only questions 4 and 7 put moves onto the plausible move list. After the opening stage, the other questions are employed in the move generation process, with questions 2, 5 and 6 being the most often used in the middle-game, and questions 5, 6, 7 and 8 during the endgame. Once there are seven moves in the plausible move table, no other moves are generated from that position. This explains why the programmers felt it important to generate the moves in a particular order — they wished to prune off most of the legal moves in every position, and preferred to do so at the move generation stage rather than use the more modern approach of generating all the moves first and then sorting them before finally discarding some.

This simple approach was used to create a tree with a maximum depth of 4-ply, and therefore a maximum of 7^4 or 2401 terminal positions requiring evaluation. In fact a further pruning mechanism was employed: a move is only put on the plausible move list if it results in an increase in score, or at least an equal score, to that which prevailed before the move was made. With a sophisticated, accurate evaluation mechanism this method might work quite well. One problem is that the program may find itself looking at a position in which no move appears to do anything but reduce the score for the side which makes it, and under these circumstances the program will select the best two moves for inclusion in the plausible move table.

Forward pruning mechanisms go a long way towards solving the problem of having the program examine too many junk variations, but to be effective without being counterproductive a forward pruning program must exhibit sound judgment when deciding which moves to prune, otherwise a superficially bad move which is really quite stunning might easily find itself eliminated from the search. It is interesting to note that the Chess Challenger machines do quite a lot of forward pruning, whereas the stronger commercially available programs do not. This in itself does not necessarily indicate that forward pruning is difficult to accomplish on a computer, and I would be interested to hear from any reader who thinks he has found a satisfactory way of pruning off a significant part of the game tree. I should remind you that Bernstein's approach of a (essentially) fixed depth search is not to be recommended. Evaluating all positions at 4-ply, irrespective of whether they are quiescent, is certain to result in feeble play unless the evaluation mechanism is sufficiently intelligent to cater for future captures in some way, as for example, Donald Michie's idea of swapoff values, which will be discussed in a future article.



The report on Bernstein's work included the moves of a game played by his program against a human opponent. I give the game here, with some comments of my own, to illustrate the standard of play that can be achieved using a primitive search process. Here, and in all future chess games, I shall employ the notation which is becoming standard for computer chess games, naming the *from* square and *to* square using conventional chess notation — White has the square a1 in his left hand corner and h1 at his right.

White: IBM 704
Black: HUMAN

1 e2-e4 e7-e5
2 f1-c4

The program had not been taught any chess openings and was playing on general principles, hence it developed a piece. It is usually a good rule to develop knights before bishops, so question 4 should have been split into (4a) Can a knight be developed? and (4b) Can a bishop be developed?

2 . . . b7-b6
3 d2-d3 g8-f6
4 c1-g5 c8-b7
5 g5-f6

Prompted by question 2, but in fact a wasted move. Firstly the bishop is more useful than Black's knight; secondly, the program ought to give more weight to developing pieces during the opening, and a move such as b1-c3 or g1-f3 is called for.

5 . . . d8-f6
6 g1-f3 c7-c6
7 e1-g1 d7-d5
8 e4-d5 c6-d5
9 c4-b5+ b8-c6
10 c2-c4?

An excellent example of why the search should not terminate at a fixed depth of 4-ply. White can win a pawn here by playing 10 f3-e5, because if Black recaptures with 10 . . . f6xe5, White will win the queen by 11 f1-e1, when Black must lose his queen for a rook. But the program would only see the variation 10 f3-e5 f6-e5 11 f1-e1 e5-e1+, and since Black is well ahead in material at this point, the program would evaluate the position as being good for black, ignoring the fact that White's next move (d1-e1) captures the black queen. Using Shannon's B strategy, accidents such as this just cannot happen.

10 . . . d5-c4
11 b5-c6+ f6-c6
12 d3-c4 e5-e4
13 f3-g5 c6-g6
14 g5-h3 e4-e3
15 f2-f3?

Here a strong move would be 15 h3-f4, attacking the black queen and preventing mate at g2, but again the program would

only have examined to the end of the four-ply continuation 15 h3-f4 e3-f2+ 16 f1-f2 g6-g2 (or b7-g2), and seen that it had lost a pawn!

15 . . . f8-c5
16 f1-e1 e8-g8

Of course e3-e2+ does not win the white queen because the pawn on e2 is pinned against the black king. But now 17 . . . e3-e2+ is a real threat.

17 b1-c3??

Which the program overlooks. There is a routine which asks "am I in check?"; but none which asks "can I give check?", and there is no question of the form "can I attack a valuable enemy piece?" As a result of these deficiencies, the program would not have put Black's next move in the top seven places on the plausible move list when considering the replies to 17 b1-c3.

17 . . . e3-e2+
18 h3-f2 b7-f3
19 g2-g3 e2-d1=Q
20 c3-d1 g6-c2
21 b2-b3 a8-d8
22 h2-h4??

Of course the program is totally lost in any case, but this move is worthy of comment because it illustrates another deficiency of forward pruning. The answers to questions 1-6 were all "no". Question 7 generated six legal pawn moves and question 8 generated piece moves, but the plausible move list was full after the first piece move was discovered, and so the program failed to spot the need to defend itself against the thread of d8-d1. Had it done so it would probably have played e1-f1, a move which requires a six-ply search to discover its refutation.

22 . . . d8-d1
23 Resigns

So with a crude search strategy and crude forward pruning, the Bernstein program was able to play recognizable chess, but extremely weakly. One important lesson that can be learned about forward pruning from the last mistake, is that your program should look further afield if the first move that it comes up with is seen to be bad. In this case, after examining the 7 chosen moves from the root of the tree, the program could see that it was losing material to 22 . . . d8-d1, but was powerless to stop it. Had it been permitted to continue its search it would have found a "better" move before too long. There is a parallel here between the drastic forward pruning method employed by Bernstein, and the iterative deepening approach used by many of today's programs. With iterative deepening, a program finds the best move it can after a 1-ply search, then it increases the depth to 2-ply and looks for a better move, then to 3-ply, and so on, until it runs out of time. Similarly, a forward pruning program should be permitted to continue its search by relaxing the pruning requirements, if it cannot find a satisfactory move early on in its search. Instead of searching 7 moves at each level, Bernstein could have examined (say) 5 moves at each level in less than one-third of the time, then when the program discovered that 21 h2-h4 and its four brothers were all dreadful moves, it could have examined all the other moves from the root of the tree, and the best five successors to each of them. This would have resulted in only a slight increase in total computation time for the move, but it would have enabled the program to see the immediate tactical consequence overlooked by the "best seven" approach. □

Bibliography

The bibliography of material on computer chess is enormous. I shall mention only a small number of particularly significant works. Further references will be given next month.

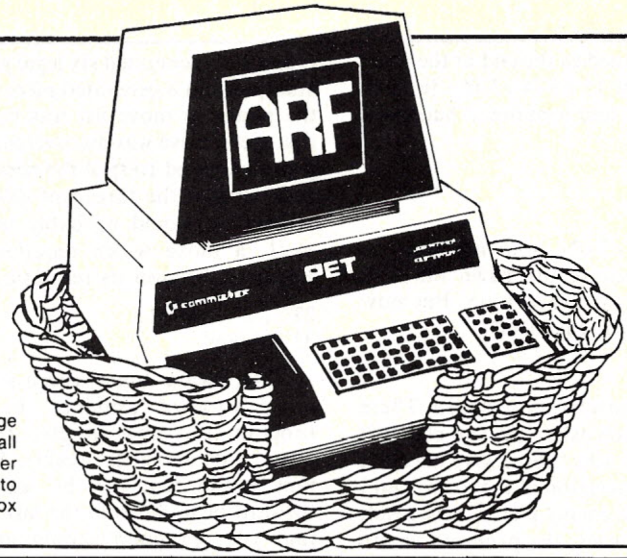
Bernstein, A., and Roberts, Michael de V. Computer v. Chess Player". *Scientific American*, Vol. 198, June 1958, pp. 96-105.

Carlson, F.R., and Zobrist, A.L. "An Advice-Taking Chess Computer". *Scientific American*, vol. 228, June 1973, pp. 92-105.

Shannon, C.E. "Programming a Computer for Playing Chess". *Philosophical Magazine*, vol. 41 (7th series) pp. 256-275.

Personal Electronic Transactions

by Gregory Yob



I am happy to hear from you, and encourage your correspondence. I will try to acknowledge all correspondence, and a SASE makes things easier for both of us. Please send your letters to "Personal Electronic Transactions" c/o PO Box 354, Palo Alto, CA 94301.

A PET Long Playing Diskette

A year ago, in the November 79 column, I looked at the Micro Technology Unlimited music board. (MTU is at PO Box 4596, Manchester, NH 03108, and the board is about $50.00). This board is a D/A converter and makes sound from programs originally written by Hal Chamberlin for the Kim.

MTU offers a software package which permits the encoding of musical selections for play on the D/A board, and Matt Ganis & Fred Covitz now have over 25 musical pieces encoded for play. (A *lot* of work!) These selections are on a PET diskette for $25.00 and give some 2½ hours playing time (Indeed, the LP Diskette). Contact Matt Ganis, Sheridan Road, RD#3, Lebanon, NJ 08833 for a copy.

If you get these songs, you will need the MTU board and the MTU music playing software. MTU now has two versions, and the one you want is the "old" software system.

Fred and Matt also offer paper listings of the encoded songs for your reference.

About Going Too Far

There's music and there's music. Though the PET can be persuaded to make sounds with pitch, envelope and timbre, there is just no match with what I can get for $7.00 at my local record store. I view the Hal Chamberlin music programs as rather a tour-de-force which extracts every last cycle out of the 6502 in an attempt to synthesize music. This is not a bad thing, except that anyone who loves music winces a bit when he hears the PET's music for the first time. The limitations are all too clear.

If the PET's 6502 were a 16 bit machine and ran 20 times as fast, the sonic quality would be superb. The issues of fidelity would then yield to those of sensitive performance. If one builds additional hardware (See the Mountain Hardware card for the Apple) the computer can be given its original job — to do the "thinking" part of performing music.

I am aware that in a few years Hal's approach to making music will be very effective. However, I am content to wait for the hardware and prefer quality later to present inadequacies. As more of us become skilled with the small machines, the "hobby" viewpoint will be replaced with a "quality" viewpoint. Keep this in mind if you are making a new product for the PET.

My PET Takes Over The World (And Blinks Some Lights)

John Bell Engineering, PO Box 338, Redwood City, CA 94064, offers a 4-channel triac board for computer control of lights and appliances which do not have motors, such as your hi-fi set. The cost is $44.95 for the fully assembled and tested board, and slightly less in kit form.

The triac is a switch for control of AC circuits, and is commonly used in light dimmer units. John Bell's unit accepts TTL logic levels and controls the triac through an opto-isolator. The PET is completely isolated from the 110V household current, so there's no chance of getting you or the PET zapped if you plug the cord in backwards.

The board comes as a PC board of 2″ by 8″ with the components mounted on 4 identical 2″ x 2″ modules. To make use of this unit, I built an enclosure with some lamp sockets as shown in Figure 1. For demonstration uses, I mounted 9 7-watt bulbs in the sockets. For more serious uses, either replace the sockets with wall-type sockets or get lamp-to-plug adapters from the hardware store.

I wired the sockets in a 1-2-3-4-1-2-3-4-1 pattern to permit some simulations of advertising signs. For example, by turning the lights to:

Light 1	*Light 2*	*Light 3*	*Light 4*
ON	OFF	OFF	OFF
OFF	ON	OFF	OFF
OFF	OFF	ON	OFF
OFF	OFF	OFF	ON

a moving pattern of one lamp on, three lamps off will move down the row of sockets.

If you look again at Figure 1, you will notice that the + side of all the isolators is set to 5 volts from the cassette port. To turn a triac "on," some current must go through the opto-isolator, which means the – side must be grounded. This means that a 1 sent to the User Port turns a lamp *off* and a 0 will turn a lamp *on.* (Note: If you make this circuit, don't try to reverse this. The 6522 I/O chip is good at pulling current to ground and poor at providing a +5 volts current. The circuit shown is reliable, and reversing things is best done in software.)

Let's work a few examples in Basic to see how a PET can blink some lights:

The first item is to set the PAØ-PA3 lines of the User Port to outputs. POKE 59459,15 takes care of this. Since the PET starts with the output register filled with zeroes, all of the lamps will light. To turn the lights off, use POKE 59471,15. This sets all of the lamp bits to "1."

Then there's a quick test:

```
FORJ=1TO100:POKE59471,Ø:POKE59471,15:NEXT
```

This flashes the lamps for a few seconds and leaves them all off.

Now, let's make the pattern mentioned earlier. That is, turn on Triac 1, then Triac 2, etc. in order, leaving only two or three lights on to give a moving effect. Here is a first attempt:

```
10 DDR=59459
20 POKE DDR,15
30 UP=59471
40 POKE UP,15
50 POKE UP,1
60 GOSUB 1000
70 POKE UP,2
80 GOSUB 1000
90 POKE UP,4
100 GOSUB 1000
110 POKE UP,8
120 GOSUB 1000:GOTO50
1000 FOR J=1TO200:NEXT
1010 RETURN
```

Lines 10 and 20 define DDR to be the User Port's data direction register and sets PAØ-PA4 to be output lines. Lines 30 and 40 define UP to be the User Port's data address and turn all of the lamps off.

Lines 50 to 120 set a value in the User Port and then call the subroutine which is simply a brief wait to let us humans see what's going on.

If you try this program, it works, but not as you expect it to. A pattern of 3 lamps

PET, cont'd. . .

on, one off will march across the triacs unit. Remembering that a "1" turns a lamp *off* provides the solution. This is an excellent excuse to use the PET's logical functions to find the numbers we need:

```
PRINT (NOT 1)AND 15
 14

PRINT (NOT 2)AND 15
 13

PRINT (NOT 4)AND 15
 11

PRINT (NOT 8)AND 15
 7
```

This works out the same as 15-1, 15-2, etc. Changing Lines 50, 70, 90 and 110 to POKE the values 14, 13, 11 and 7 respectively will now make the program work correctly.

A minor problem with the program is that the lamps stay on when you press the STOP key. Changing the subroutine makes this more graceful:

```
1000 FOR J=1 TO 200:NEXT
1010 GET A$:IF A$="" THEN RETURN
1020 POKE UP,15
1030 END
```

Now a press of any key stops the program and turns off the lights.

As an exercise, I challenge you to make these patterns on the lamps:

```
●●OO●●OO●    ●●OO●●OO●
O●●OO●●OO    ●OO●●OO●●
OO●●OO●●O    OO●●OO●●O
●OO●●OO●●    O●●OO●●OO
   (1)          (2)

●O●O●O●O●    ●OOO●OOO●
O●O●O●O●O    ●●OO●●OO●
●●OO●●OO●    ●●●O●●●O●
OO●●OO●●O    ●●●●●●●●●
   (3)          (4)
```

The triacs are each capable of handling up to 600 watts. One practical application is to use the PET to handle the lamps in the house when you are away. Use of the TI$ variable and RND for some changes of pattern can be effective for fooling burglars.

However, there's a much nicer idea — If you order the triacs immediately there's enough time to get ready for Christmas. Imagine your Xmas tree with four strings of lamps controlled by the PET instead of those boring flasher units!

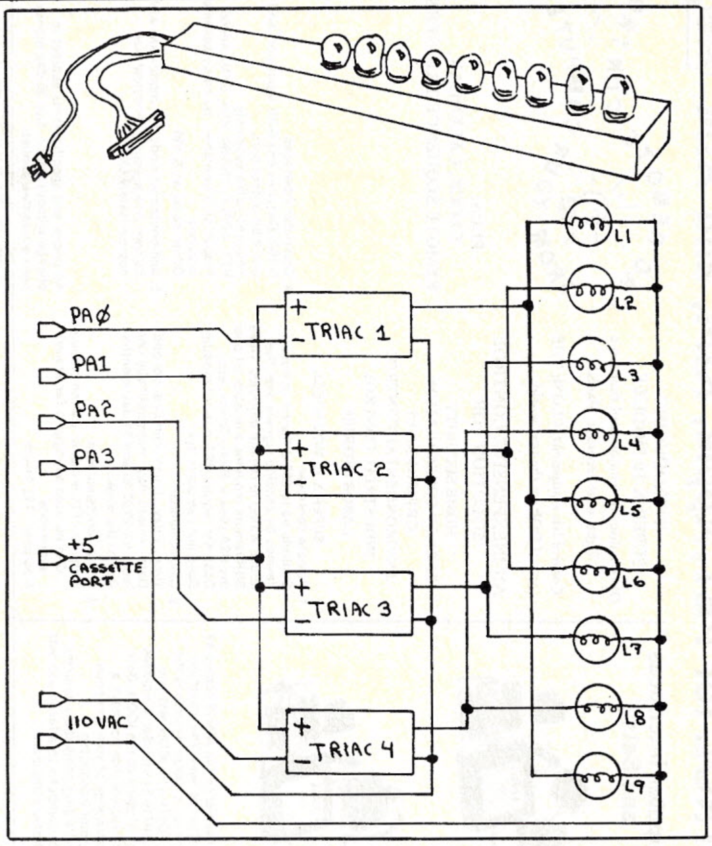


Figure 1.

The upper drawing shows the completed triac enclosure with some low-wattage lamps installed. The enclosure is about 30" long by 2" wide by 3" wide. The Bell Engineering board fits in the left side which has no lamp sockets. The parts cost of this unit is about $25.00.

The lower diagram shows how the triacs are connected to the PET and the lamps. The User Port Lines PA0 through PA3 control Triacs 1 through 4. The +5 supply is taken from the Cassette Port next to the User Port. These lines are:

2 or B	from Cassette Port. (Check with a VOM)
C,D,E,F	from User Port.

If you trace the circuit, Lamps L1, L5 and L9 are controlled by Triac 1, L2 & L6 by Triac 2, L3 & L7 by Triac 3 and L4 & L8 by Triac 4. This arrangement permits sequencing of the lights in a pattern which moves along the row of sockets.

There are Indeed Some Women

I have a few replies to my request for letters from women, and the one reproduced below really tells quite a bit of the differences between men and women when it comes to computers. Take heed!

Dear Gregory:

I thought your February question about women computer freaks deserved some sort of answer (I just got my copy back from the local computer freak)!

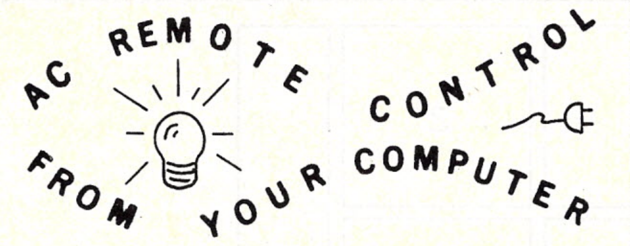
I'm a commercial programmer working for a consulting company. At my current assignment there are hundreds of programmers, nearly half women. There are several computer freaks, but not a great many. I know one of them rather well, but he is a very unusual person in many ways — for one thing, he's really a system programmer who happened to get hired in applications. He's also single. Most of the people here, not just the women, have things to do when they get home. When they do have leisure time, they generally prefer not to do computing. As for those people who are not programmers, you may have noticed that high-technology hobbies are usually practiced by men — such activities as assembling electronic kits, fixing cars, and building model planes attract a few women, but mostly men.

Myself, I study music and play with two groups; I'm renovating a brownstone, I garden, do crafts and needlework, and belong to various clubs. I do my computing at work, where I get to play lots of games, from Office Politics to UNIX trial.

It occurs to me that a lot of personal computing involves games like Space Invaders and Adventure. You may have noticed that women very seldom play these games. I like them even less than most women do; I find them aggravating and a waste of time.

As a women programmer, I am interested in applications, however pedestrian, that fill a need, and in the human/machine interface. I would be interested in doing some computing for the various voluntary organizations I belong to — except that I think their computing should be centralized (shared) rather than distributed, to avoid waste of resources. Mailing lists, hotline sources, and a musician's directory may not be sexy, but there is a need. I just can't get into a better WUMPUS.

Very truly yours,
Sandra Greer
Brooklyn, NY



TRS-80 PET S100
APPLE KIM AIM65

INEXPENSIVE CONTROL SOLUTION FOR
HOME SECURITY • ENERGY CONSERVATION
GREENHOUSES • ENVIRONMENTAL CONTROL
INDUSTRIAL CONTROL • LABORATORIES

CmC's μDAC system now includes an interface to the BSR X-10 remote control modules. These low-cost modules allow control over lamps, motors and appliances. With the CmC X-10 interface your computer can control 256 separate devices. Lamps can be turned on or off, dimmed or brightened. Alarms, kitchen appliances, hi-fis, TVs, motors, pumps, heaters and more can be put under your computer's control.

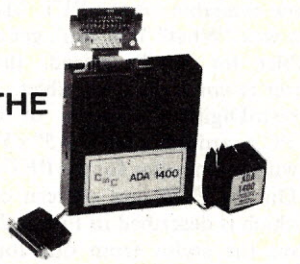
Direct plug-in and software for most computers.

Circle the reader service number, call or write for our latest catalog.

CmC **CONNECTICUT microCOMPUTER, Inc.**
150 POCONO ROAD
BROOKFIELD, CONNECTICUT 06804
TEL: (203) 775-9659 TWX: 710-456-0052

PET to NEC and CENTRONICS PRINTER ADAPTER

LOWEST COST COMPLETE INTERFACE ON THE MARKET



Simple to use — low cost — designed for NEC 5530 Spinwriter and Centronics parallel printers. Works with WORDPRO and other software. Switch for upper-lower case conversion or upper case only.

Plugs into the PET and into the printer — all cables and connectors included — extra IEEE connector for Commodore disk drives.

Uses BASIC PRINT statements — no machine code needed.

Device address selectable — works with other peripherals.

$129 complete — compare to others at $225.

Generous dealer discounts.

Assembled and tested. Our usual 30 day money back trial period applies.

Order direct or contact your local computer store.

VISA and M/C accepted — send account number, expiration date and sign order.
Add $3. per order for shipping and handling — foreign orders add 10% for air postage.
Mention this magazine with your order and deduct 2%.

CmC **Connecticut microComputer, Inc.**
34 Del Mar Drive, Brookfield, CT 06804
203 775-4595 TWX: 710 456-0052

• PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET

PET PRODUCTS

Programs — Workbooks
for Floppy Disk — for Cassette

See your Dealer

Dealer Inquiries Invited

Put your PET to work!

T I S

PROGRAMS		WORKBOOKS		
SW-1*	MAILB mailing list system	WB-1	Getting Started with Your PET	$3.95
SW-2*	CHECKBOOK record	WB-2	PET String and Array Handling	$3.95
SW-3*	ACCOUNTS keep track of who owes you how much	WB-3	PET Graphics	$4.95
SW-4	MEDIT create and maintain data files	WB-4	PET Cassette I/O	$4.95
SW-5*	CALENDAR appointments, meetings at a glance	WB-5	Miscellaneous PET Features	$3.95
		WB-6	PET Control and Logic	$3.95

*These programs are special purpose data base management systems. They all can:

- Sort numeric or string fields
- Select based on numeric or string (=)
- Select based on substring match
- Select based on range of entry number

Prices: $9.90 each for programs using cassette storage for data
$12.95 each using sequential floppy disk storage for data.
Price Includes 40 - 60 page instruction manual

Add $2.00 for shipping and handling

Money back guarantee

TIS
P.O. Box 921, Dept. CC
Los Alamos, NM 87544

VISA

On bank card orders, give all numbers

PET is a trademark of Commodore Business Machines

• PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET • PET

CIRCLE 193 ON READER SERVICE CARD

KRELL SOFTWARE

presents for the TRS-80 PET, Apple II, and Apple II Plus

COLLEGE BOARDS

The best way to sharpen your skills for the College Board SAT Exams is to work on actual examinations. Each of these 4 programs confronts the user with a virtually limitless series of questions and answers. Each is based on past SAT exams and presents material of the same level of difficulty and in the same form as used in the verbal and mathematical portions of the College Board Examinations. Scoring on each exam is provided in accordance with the formula used by College Boards.

COLLEGE BOARD—VOCABULARY	19.95
COLLEGE BOARD—WORD RELATIONSHIPS	19.95
COLLEGE BOARD—MATH PART A	19.95
COLLEGE BOARD—MATH PART B	19.95
Complete Set	59.95

TIME TRAVELER

The best of the adventure games. Confronts player with complex decision situations and at times, the demand for real time action. Using the time machine, players must face a challenging series of environments that include: the Athens of Pericles, Imperial Rome, Nebuchadnezzar's Babylon, Ikhnaton's Egypt, Jerusalem at the time of the crucifixion, the Crusades, Machiavelli's Italy, the French Revolution, the American Revolution and the English Civil War. Deal with Hitler's Third Reich, the Vikings, etc. Involve yourself with historical military and government operations, markets, etc. in fascinating game situations. Each game is unique! $24.95

THE SWORD OF ZEDEK

Fight to overthrow Ra, the Master of Evil. In this incredible adventure game you must confront a host of creatures, natural and supernatural. To liberate the Kingdom, alliances must be forged and treasures sought. Treachery, deceit and witchcraft must be faced in your struggle as you encounter wolves, dwarves, elves, dragons, bears, owl, orcs, giant bats, trolls, etc. Each game is unique in this spectacular and complex world of fantasy. $24.95

*All programs require 16K • TRS-80 Programs require Level II BASIC. • Apple programs require Applesoft BASIC.

Send check or money order to **Krell Software**
21 Millbrook Drive, Stony Brook, NY 11790 (516) 751-5139

CIRCLE 215 ON READER SERVICE CARD

PET, cont'd. . .

The PrestoDigitizer™ Tablet

An unusual graphics entry device, the PrestoDigitizer (Contact: Innovision, PO Box 1317, Los Altos, CA 94022. Price: $50.00) is now available for the PET. My first reaction was "What? You can't get a graphics tablet for $50!" Indeed, the PrestoDigitizer is *not* a graphics tablet.

The PrestoDigitizer consists of a stylus and a PC board mounted in a 5" x 5" plastic base with a connector for the PET's User Port. On the PCB is a pattern of copper foil which is described in Figure 2. As you move the stylus from one foil region to another, the PET will see a changing pattern on the User Port bits, and with appropriate software the digits 0-9 and letters A-Z may be recognized. For example, the sequence: 1 2 4 5 6 7 may be used to recognize the letter Z. (In most PrestoDigitizer programs, touching Region 7 on the right signifies the end of a letter.)

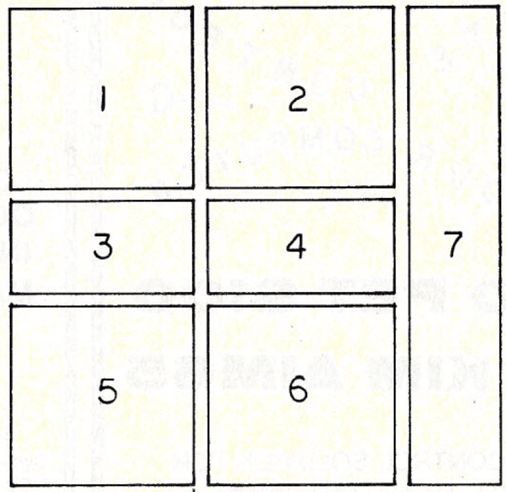


Figure 2. **The PrestoDigitizer Pad**

Seven copper foil regions are placed in a 3" x 3" square on a printed circuit card. Each of the regions (except #4) are connected to the PET's User Port (PA0-PA5). By grounding a region with the stylus, the PET detects a contact and a series of contacts can be used to define a letter. For example, the series 1-2-4-5-6 can define the letter "Z." Region #7 is normally used to indicate "end of character."

My second reaction was, "Hey, wait a minute! Something this limited and simple isn't worth $50.00." Well, after a look and a bit of thought I conclude the PrestoDigitizer *is* worth $50.00 for some people. If you are handy with a keyboard, the PrestoDigitizer will be only a toy. If you don't get along well with keyboards, this gadget will be very worthwhile.

The documentation for the PrestoDigitizer is excellent. I received descriptions of three programs (included on a cassette) which indicate some starting uses for the PrestoDigitizer, a reprint of an article in *Recreational Computing* describing the principles of the PrestoDigitizer in detail, and several pages devoted to how to use the PD in your own programs.

Two of the programs, QUIZZER and HANDWRITING, have a "learn" mode in which your own patterns of motion for the digits and letters are entered into the program. I do have a small complaint on these, that on some digits and letters I made an error, and there was no obvious way to re-enter a letter's pattern. One simple method is to have a short dialog like:

I NOW HAVE THE LETTER D.
PLEASE RE-ENTER THE D.
THANK YOU. I NOW KNOW WHAT D LOOKS LIKE.

Two successive entries of the same letter pattern would "memorize" the pattern.

In learning an alphabet, the PrestoDigitizer strikes a nice balance between your normal variations in handwriting and the simplicity of the device. The only meaningful thing is the sequence of regions contacted by the stylus. The time you take, or lifting the stylus from the pad makes no difference.

Before you dismiss the PD as a toy, bear in mind that children, non-typists just learning computers, and persons with motor handicaps of the hand will find it a valuable tool, at a very reasonable price. When the PD is in "learn" mode, *any* sequence is learnable as *any* letter. All that is needed is that each letter have a unique sequence.

If any of you out there would like to realize the PrestoDigitizer's recognition and learning programs in machine language, I am most interested. Innovision would like to see such a package, especially one that permits the use of a PD in place of a keyboard, and I know one person who could use this. Such a program would put the entered letters into the PET's keyboard buffer to let the full capabilities of the PET be used. (My friend programs the PET, though very laboriously.)

Machine Language Goodies #3, 4, 5 & 6

In some video games you are driving a car along a roadway. The car remains in the center of the screen and the road moves down the screen. I thought it would be fun to do a similar thing with the PET's screen by moving the *entire* screen up, down, left or right with wraparound. Characters going off one edge of the screen will appear on the opposite edge.

Since this means moving some 1000 characters around, I wrote these functions in machine language to be called by the SYS function. You don't want to wait 30 seconds to shift the screen one position.

I then wrote a small "drive-it-around" program to illustrate how these subroutines may be used. Figure 3 shows the HEART DRIVER program as listed on my printer. (There's far too many DATA statements to attempt typing via my Selectric without errors. Unfortunately, it takes only *one* error to smash machine language!)

Before looking at the program in detail, some of the listing rules need to be explained. If you look at Line 100, you will see the phrase ROUTINEsp LOADED. The "sp" indicates that a blank went between the two words. On Line 170, the letters M, B, and N are preceded by the tilde (~) mark. This means that the graphics characters Shift-M, Shift-B, and Shift-N are used here. If you make the substitutions, the familiar numberpad-moves diagram will appear. Note that blanks, i.e., between the "sp"s do *not* count. To be precise, blanks in a program line are in the program, except inside quote marks. *All* blanks in quotes are replaced by "sp"s.

Cursor movements are indicated by "up", "dn", "lft", "rt" for Cursor Up, Cursor Down, Cursor Left, and Cursor Right, respectively. In Line 530 is a "rvs" which stands for the Reverse Field key.

So, how does this thing work? Lines 10 through 100 read the DATA statements and put the 6502 code into the first and second cassette buffers. (Take heed! No Tape I/O permitted!). Unfortunately I couldn't fit it all into the second cassette buffer.

Line 110 is a delay to show the ROUTINE LOADED message, and Line 115 eats up any characters in the input buffer. The program returns to Line 115, and up to 9 characters may be lurking to be GETted. Lines 120 to 230 give instructions and wait for a keypress to begin.

Line 240 defines where the "currently down key" location is. For "old" PETs this is 515, and "new" PETS use 151. Location 50003 is a zero for "old" PETs and a one for "new" PETs. This is a convenient way to make programs run on both model ROMs. (Thanks to Len Lindsay for this tip. See the July 79 *Kilobaud,* Pg 72 for more regarding the "new" and "old" ROMs.)

Line 250 clears the screen. Lines 260 to 270 place grouips of Shift-Q (solid balls)

on the screen. I don't have room to explain the method, so it is a puzzle for you to figure out. Line 275 puts the heart symbol on last as the goal to reach. In 280, the cursor is moved to the center of the screen and the diamond printed to indicate the car. That's 12 down and 19 to the right from the home position. Line 290 defines the SYS calls for moving the screen. These are:

863	U	Move Screen Up One Unit
905	D	Move Screen Down
793	R	Move Screen Right
827	L	Move Screen Left

If you RUN the program and press the STOP key, try SYS to each of these and see what happens. Then try a loop like FOR J=1 TO 40: SYS 793:NEXT

C is defined to be the center of the screen, which is where the car is positioned. The time is noted in Line 305 and Line 310 checks for any key down. (255 is the no keys pressed value.) The PET uses some odd codes to show which key is down, and the values used here are:

58 (7 key)	50 (8 key)	57 (9 key)
42 (4 key)	34 (5 key)	41 (6 key)
26 (1 key)	18 (2 key)	25 (3 key)

I am too short of space to explain how you find these values — a topic for another time. Lines 320 to 410 print a "+" to show where the car was, and then move the screen in the direction opposite to the car's intended motion. This gives the illusion of moving the correct direction. Notice that the diagonal moves are done by two SYS to move Up or Down and then Right or Left. Line 410 corrects for non-directional keys being pressed by restoring the car's image.

Once the move is made, we check for a collision by looking at the screen's center to see what's there. A space or a + is checked for in Line 502, and the heart is looked after in Line 505. (The PET's POKE codes to the screen are *not* the ASC values for the letters. Yet *another* topic!) The game ends in Line 530 with a time taken message and a jump back to the start at Line 115. A collision with a dot produces some random moves in Lines 540 to 600, and then a delay. We again check for a dot, and repeat the collision process until the car lands on a non-dot. Try a crash — it's kind of fun!

I leave you this month with a challenge and promise. Using the SYS calls for moving via the 7, 8 and 9 keys only, do a driving along a road game with the road boundaries scrolling *down* the screen. The catch is that the road is a lot longer than 25 lines. For starters, make a "track" that is 200 lines long, or about 8 screens in length. How about a few potholes, ruts, and oil slicks for fun? Note I do *not* want to see a randomly created track, but one that repeats after a while. The promise? I will publish your program here. □

Figure 3. **Heart Driver Program**

The listing conventions of this program are described in the text.

```
HEART DRIVER

10  AD=690
20  READ B:IFB<0 THEN 100
30  POKE AD,B:AD=AD+1:GOTO20
100  PRINT"ROUTINEsp LOADED"
110  FORJ=1TO500:NEXT
115  GETA$:IFA$>""THEN115
120  PRINT"clr WOULDsp YOUsp LIKEsp
       TOsp GOsp FORsp Asp SPIN?
130  PRINT:PRINT"JUSTsp PRESSsp THEsp NUMBERsp KEYS
140  PRINT"TOsp DRIVEsp YOURsp CAR.....
150  PRINT
160  PRINT"sp sp sp 7sp sp 8sp sp 9
170  PRINT"sp sp sp sp ~Msp ~Bsp ~N
180  PRINT"sp sp sp 4~Csp 5sp ~C6
190  PRINT"sp sp sp sp ~Nsp ~Bsp ~M
200  PRINT"sp sp sp 1sp sp 2sp sp 3
210  PRINT
212  PRINT"YOURsp GOALsp ISsp THEsp ~Ssp SPOT
214  PRINT"GOODsp LUCK.....
216  PRINT
220  PRINT"PRESSsp ANYsp KEYsp TOsp BEGIN...
230  GETA$:IFA$=""THEN230
240  PK=515:IFPEEK(50003)THENPK=151
250  PRINT"clr ";
260  FOR J=1TO20+15*RND(1)
262  W=32768+1000*RND(1)
263  FOR K=1TO10+15*RND(1)
264  ON1+4*RND(1)GOTO265,266,267,268
265  W=W+1:GOTO269
266  W=W-1:GOTO269
267  W=W+40:GOTO269
268  W=W-40:GOTO269
269  IFW<32767ORW>33767THEN262
270  POKE W,81:NEXTK,J
275  POKE32768+1000*RND(1),83
280  PRINT"dn dn dn dn dn dn dn dn dn dn dn dn
       rt rt rt rt rt rt rt rt rt rt rt
       rt rt rt rt rt rt rt rt ~Zlft ";
290  U=863:D=905:R=793:L=827:C=33267

300  REM LOOK FOR MOVE
305  T=TI
310  Z=PEEK(PK):IFZ=255THEN310
320  IFZ=34THEN310
330  PRINT"+lft ";
340  IFZ=50THEN SYSD:GOTO500
350  IFZ=18THEN SYSU:GOTO500
360  IFZ=41THEN SYSL:GOTO500
370  IFZ=42THEN SYSR:GOTO500
380  IFZ=58THEN SYSD:SYSR:GOTO500
390  IFZ=57THEN SYSD:SYSL:GOTO500
400  IFZ=26THEN SYSU:SYSR:GOTO500
410  IFZ=25THEN SYSU:SYSL:GOTO500
420  PRINT"~Zlft ";:GOTO310
490  REM CHECK FOR COLLISION
500  Q=PEEK(C)
502  IFQ=20 OR Q=43 THEN 420
505  IFQ=83 THEN 530
510  IFQ=81THEN540
515  GOTO420
520  REM COLLISION FOUND
530  PRINT"rvs lft lft lft lft lft lft
       lft lft lft lft GOTsp THEREsp IN"
       INT((TI-T)/60)"lft sp SECONDS"
535  FORJ=1TO2000:NEXTJ:GOTO115
540  PRINT"~Qlft ";
550  ON 1+4*RND(1) GOTO 560,570,580,590
560  SYSU:GOTO600
570  SYSD:GOTO600
580  SYSR:GOTO600
590  SYSL:GOTO600
600  IFRND(1)>.3THEN550
610  FORJ=1TO500:NEXTJ
620  GOTO 500

1000  DATA 255,127,0,128,39,128,136
1010  DATA 2,151,131,191,131,162,11
1020  DATA 181,10,157,122,2,202,16
1030  DATA 248,162,11,189,178,2,149
1040  DATA 10,202,16,248,96,162,11
1050  DATA 189,122,2,149,10,202,16
1060  DATA 248,96,216,24,165,10,105
1070  DATA 40,133,10,144,2,230,11
1080  DATA 24,165,12,105,40,133,12
1090  DATA 144,2,230,13,24,165,14
1100  DATA 105,40,133,14,144,2,230
1110  DATA 15,96,216,56,165,18,233
1120  DATA 40,133,18,176,2,198,19
1130  DATA 56,165,20,233,40,133,20
1140  DATA 176,2,198,21,96,32,190
1150  DATA 2,162,25,32,43,3,32
1160  DATA 222,2,202,208,247,32,211
1170  DATA 2,96,160,39,177,12,72
1180  DATA 177,10,145,12,136,208,249
1190  DATA 104,145,12,96,32,190,2
1200  DATA 162,25,32,77,3,32,222
1210  DATA 2,202,208,247,32,211,2
1220  DATA 96,160,1,177,10,72,177
1230  DATA 12,145,10,200,192,40,208
1240  DATA 247,104,145,10,96,32,190
1250  DATA 2,162,24,160,40,177,10
1260  DATA 145,16,136,208,249,160,40
1270  DATA 177,14,145,10,136,208,249
1280  DATA 32,222,2,202,208,241,160
1290  DATA 40,177,16,145,10,136,208
1300  DATA 249,32,211,2,96,32,190
1310  DATA 2,162,24,160,40,177,20
1320  DATA 145,16,136,208,249,160,40
1330  DATA 177,18,145,20,136,208,249
1340  DATA 32,1,3,202,208,241,160
1350  DATA 40,177,16,145,20,136,208
1360  DATA 249,32,211,2,96
1370  DATA -1
```

Outpost: Atari

George Blank

Report from the Forward Observer

Atari is working on a Pascal for their computers, scheduled to be released in the first half of 1980 if all goes right. Their Pascal will not be USCD Pascal, but will have a number of differences. It will compile into either P-Code or 6502 machine language.

Disk Operating System 2 should be ready about the time this column appears. The primary problem with DOS 1 is that random files do not work, although the new DOS is significantly faster and uses less operating memory. The memory is saved by using overlay methods, keeping some of the DOS out on the disk unless it is needed. The new DOS will cost present disk owners $25 (Part number CX 8104). There is another problem involved. The speedup of the DOS requires a different formatting routine. Unfortunately, the old formatting routine is in ROM, so the only way to take advantage of the higher speed is to buy formatted disks direct from Atari. The scheduled price is $25 for 5 (Part number CX 8110).

Another scheduled Atari project is a mailing to all registered Atari owners containing information on software publishers and others supporting the Atari computers. With projects like this and Atari releasing documentation to warranty card registrants, it is important to return your registration card. If they don't have your address, they can't send you anything!

Atari has been negotiating with Microsoft for a new Atari Basic. If things go smoothly, look for the new Basic in mid 1981.

SoftSide magazine began to cover the Atari computer with their August issue by printing three complete games and several articles. Regular coverage of the Atari is promised. More information should be available in their ad elsewhere in this issue.

George Blank, Foster Road, Milford, NH 03055.

They Said it Couldn't be Done!

My taste in poetry can be decidedly lowbrow, but I have always liked this little ditty:

They said it couldn't be done
So he went right to it.
He took that thing that couldn't
be done . . .
And couldn't do it.

James Garon of *SoftSide* has a different tale to tell. According to the Atari manual, graphics 8 gives you only two shades of one color. James discovered that when you draw a vertical line in an odd numbered column, it is a different color from a vertical line in an even numbered column, and drawing in both columns gives still another color. Here is a little program to demonstrate the method:

```
10 GRAPHICS 8
20 COLOR 1
30 FOR X = 1 TO 319 STEP 2
40 PLOT X,0 : DRAWTO X,40
50 NEXT X
60 FOR X = 0 TO 318 STEP 2
70 PLOT X,41 : DRAWTO X,80
80 NEXT X
90 FOR X = 0 TO 319
100 PLOT X,81 : DRAWTO X,120
110 NEXT X
120 PRINT"IMPOSSIBLE?"
130 GOTO 130
```

According to the friendly experts at Atari, if you use assembly language it is possible to get 128 colors on the screen at the same time in high resolution.

Visicalc Update

Visicalc should be ready from Personal Software by the time this column appears, or shortly thereafter. Since Doug Green wrote an excellent review in the August issue of this magazine, I will not review it, but simply report that I have been using a preliminary copy with the instructions from the Apple version, and it is the same program.

Visicalc is the first Atari program I have encountered with its own disk operating system. You use it without a ROM cartridge, and it boots and loads automatically from the disk.

Intelligence Report

Four of the things that make the Atari computers special are custom integrated circuits built into the computer. These four chips go by the names PIA, ANTIC, CTIA, and POKEY. Many of the features that set the Atari apart from earlier and more primitive personal computers are located in this hardware.

The PIA chip controls information going to and from the joysticks, paddles, or controller jacks and also handles interrupt requests. There are two standard peripheral interface adaptors (That is what PIA means). In last month's column I described how to use these ports.

The ANTIC chip controls direct memory access for fast graphics and transfers of information, the non maskable interrupts, vertical and horizontal scrolling of your TV set, and contains the position registers for the light pen.

The CTIA chip offers priority control so that objects can overlap, such as the basketball players in Atari Basketball. It can also control up to 4 "players" and 4 "missiles". A player is an object that can be displayed on the screen which is a maximum of eight bits (or graphics blocks) wide. A missile is an object that is no more than 2 bits (blocks) wide. There is no limit on height. The four missiles can be combined into a fifth player, as is done in the basketball game to form the ball. This chip also maintains the colors and luminances of the objects, detects collisions between the objects, and controls their horizontal position. Yet another function of this busy chip is to monitor the keyboard switches and paddle and joystick triggers.

The POKEY chip is responsible for scanning the keyboard, controlling the serial port to the printer, disk drives, and

cassette recorder, converting the position of the eight paddle controls to a number the computer can read, creating the sound for the four audio channels, updating the internal timers, and random number generation.

The actual use of these chips for fancy graphics is too complex for this column. However, those of you who seek more information can find a description of ANTIC graphics in the August 1980 issue of *Byte,* written by Chris Crawford and Lane Winter of Atari. Beware; the method is much too tough for beginners.

Programming Feature

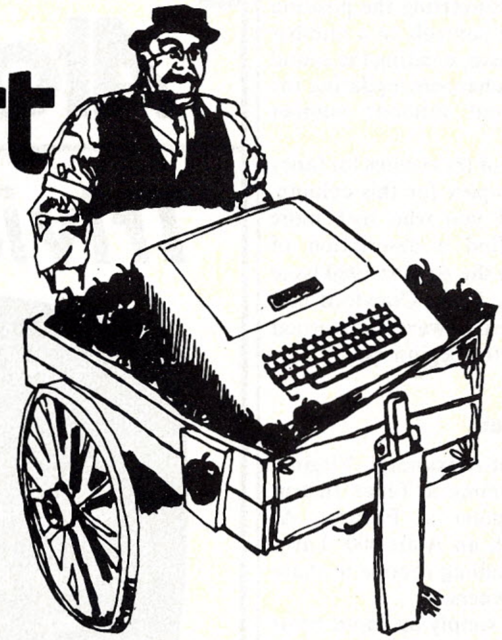
Larry Seftor of Alexandria, Virginia works by programming a Texas Instruments ASC Computer in Fortran. At home he plays with an Atari 400. Larry sent in an error handling routine to share with other Atari owners.

If the error is simply an input error (error code 8), the program returns to the same line and tries again. Otherwise, it prints out the error number, the line the number is in, and then lists the line for editing. This basic technique can be expanded to handle many other kinds of errors as you seek to make your programs fail safe. If you are maintaining a glossary of computer terms, you should know that this process goes by the esoteric and highly technical term of "Idiot proofing". □

```
10 TRAP 1000 (First line of program)
-
       (Your program fits here)
-
1000 TRAP 1000 : ET = PEEK(195)
     * 256 + PEEK(186)
1010 EL = PEEK(187)
1020 IF ET <> 8 THEN PRINT : PRINT"
     ERROR ";ET;
     " IN THE FOLLOWING LINE:"
     : LIST EL : STOP
1030 GOTO EL
```

"Computer Room? Switch to bad mood tapes and send out nasty letters." ©Creative Computing

Apple-Cart



Chuck Carpenter

Correspondence is always welcome and a response will be made to those accompanied by a SASE. Send your letters to: Chuck Carpenter, 2228 Montclair Pl., Carrollton, TX 75006.

In the May '80 issue, the Apple Cart column included a section on assembly language fundamentals. To continue with additional fundamentals, this column will describe the principals of indexing. Another 6502 microprocessor feature includes the use of interrupts. The second part of this column will describe simple interrupt control hardware (to monitor remote switches) and a machine language program featuring indexing. The circuit and program will let you monitor the "outside world" and run your favorite program at the same time.

Indexing Principals

Sequencing a data table to print character strings is easily handled in 6502 assembly language by "indexing" the table. Apple's 6502 microprocessor includes a variety of indexing instructions. Two of these, absolute indexing and one form of indirect indexing will be described here. Once the principals of indexing are understood, you can easily master each of the 6502 indexing modes. The examples included will help you gain this understanding.

Absolute Indexing

Absolute indexing is accomplished by locating the characters in a table relative to the starting address of the table. To determine the relative position in the table, a displacement value is added to the starting address. In the 6502, there are 3 registers used for processing data in a program. One is the accumulator or A register and the others are the X and Y index registers. Absolute indexing uses the A register — to contain the base address — and the X or Y register to hold the offset or index value. In our examples, we'll use the X register. This sounds confusing so let's look at an example.

In this program line example, the main program starts at address $0800 (the $ symbol means a hexadecimal number). The assembled machine language represents the indexing opcode and starting address (operand) of the table. Remember that the operand determines the address mode which modifies the instruction and establishes the final opcode. This operand indicates an indexing operation; address $0900 plus the current value of the X register. The character table starts at address $0900. Each character in the table will be found at the absolute value of X added to the base address $0900. The value of X is the displacement value.

```
         MACHINE
ADDRESS  LANGUAGE   LABEL  INST  OPERAND      COMMENT
-------  --------   -----  ----  -------      -------
0800-    XX XX XX   INDEX  LDA   $0900,X      Index Table

Opcode
Address

Opcode

Operand

Symbolic Label

Mnemonic Instruction

Table Address +
Character displacement
```

Another View Point

One more example will help show the mechanism of absolute indexing. Let's examine a program segment that will display a character string. For instance, if you wanted to display your name and the year, you might set up a table as shown in Figure 1. First, you need to start with the offset value for the table in the X register. In this case we start with zero. The first character is at address $0900. So, we don't need a displacement for the first character. Then we load the accumulator with the character found at address $0900 plus the current value in the X register. The operand $0900, X indicates this condition. That is, the accumulator is to hold the character at base address $0900 indexed absolute by the value of the X register. The first time through, the character loaded in the accumulator will be the first letter of your name. This character is printed out on the screen by the monitor routine at address SFDED. Next the X value is compared to the value for the end of the list + one. Because the X register and the end-of-list value are not equal, the routine branches back to the label INDEX to get the next character in the table.

The second time through, the value in the X register is now $01 and the accumulator will be loaded with the character in $0901. The print-out, incre-

SUPER-TEXT II™

From MUSE, the Leader in Quality Software

With its easy to use key sequences, the ST II **Word Processing** system provides fast editing of letters, manuscripts, books, or other printed material. Automatic column totals, column alignment, and on screen tabbing eliminate tedious formatting and make **Financial and Summary Reporting** a snap. **Data Base Management** is simplified by the most advanced file handling system and makes a world of information instantly available ($150.00).

Completely new documentation designed for first time users and a backup copy of the program disk is included with every program.

TRADE-INS!!

Trade in your present word processor for Super-Text II. Send documentation manual, and proof of purchase.

Trade-in Value:

Super-Text	$100.00
Easy Writer	$50.00
Apple Writer	$50.00
Apple Pie	$50.00

SUPER-TEXT ADD-ON MODULES
For Even More Versatility

Personalize and customize form letters with the ultimate letter writer, the Super-Text **Form Letter Module** ($100.00). Use it with the **Address Book Mailing List Program** ($49.95) or by itself to end repetitious composition of letters, contracts, leases, and other documents. Add the Super-Text **Terminal Module** ($75.00) and a D. C. Hayes Micromodem to communicate with any computer in the world.

Coming soon . . . The Super-Text Virtual Memory Assembler—the MUSE in-house development system.

Apple II or Apple II Plus (48K).

For a free catalog or the name of your nearest MUSE Dealer, contact:

MUSE SOFTWARE™

330 N. CHARLES STREET
BALTIMORE, MD 21201
(301) 659-7212

CIRCLE 226 ON READER SERVICE CARD

```
LABEL  INST  OPERAND     COMMENT
-----  ----  -------     -------
       LDX   #$00      ; index displacement
INDEX  LDA   $0900,X   ; read table
       JSR   $FDED     ; print character
       INX             ; next character
       CPX   #$0E      ; table length + 1
       BNE   INDEX     ; back if not done
       RTS             ; end it if it is

0900- Y
0901- O
0902- U
0903- R
0904- spc
0905- N
0906- A
0907- M
0908- E
0909- spc
090a- Y
090b- E
090c- A
090d- R
```

Figure 1.

Put as many characters as needed in the table. Use the ASCII value as shown on pages 138 and 139 of the Applesoft reference manual.

```
LABEL  INST  OPERAND     COMMENT
-----  ----  -------     -------
       LDX   #$04      ; 5 char. offset
INDEX  LDA   $0900,X   ; index table
       JSR   $FDED     ; print character
       DEX             ; next character
       BPL   INDEX     ; back it not done
       RTS             ; end it if it is

0900- 4F  (O)
0901- 4C  (L)
0902- 4C  (L)
0903- 45  (E)
0904- 48  (H)
```

Figure 2.

Short-table indexing.

menting, and comparing operations are carried out as before. The cycle is then repeated until the entire table is completed. When the compare is equal, the routine ends. Note that the characters all include the high-order bit. That is, a hex 80 is added to the ASCII value of the character. Otherwise, because of an Apple video characteristic, the output would be in reverse video. Indexing in this manner will allow up to 256 characters in a table. (Decrementing from $FF to $00 equals 256 steps.)

Try It This Way Too

To index a table longer than 256 bytes, you would need to use other techniques. One way would be to hold the index displacement constant and increment the memory locations. An example will be included with indirect indexing later. For shorter tables — less than 128 characters — a shorter program is possible. An example is shown in Figure 2. This version is similar to our previous example. Except, it's shorter and reversed.

The displacement value starts at the end of the table and the last character is read first. Rather than increment X we decrement it. And, the branch to get the next character is taken as long as X remains plus. Plus (or minus) is determined by the sign bit of the placement value. If the eighth bit is a zero, the value is plus. If the eighth is a one, the value is minus ($00 to $7F are plus — $80 to $FF are minus). When the value of X is decremented from $00 to $FF the sign bit becomes minus and the program ends. Because half the indexing values will be plus and the other half minus, this technique will only allow a 128 character table to be indexed.

This is the simplest form of indexing. By incrementing or decrementing the index register you can sequentially "pick" the data from your table. The operand (memory location plus the value in the index register) points to the character in the table.

Indirect Addressing

Indirect addressing does essentially the same thing as absolute except one more step is added. With indirect addressing, the operand, plus the index value, points to the memory location that points to memory where the table is. Simple, right! Here's a diagram to illustrate the technique.

```
0800-       LDY #$00
0802-       LDA ($0300),Y This operand

0805-       INY
  .
  .
  .
0300-00     0900+Y       Points to this
0301-09                  Memory location
  .
  .
  .
0900- 54 (T)             Which points to
0901- 41 (A)             the table
0902- 42 (B)
0903- 4C (L)
0904- 45 (E)
```

For simple table-reading programs, absolute indexing is adequate. Indirect indexing is more appropriate where code economy and speed of operation are important. For such applications, you must index from page zero. In the example above, the code is for pages 3, 8 and 9 (arbitrary for purposes of illustration). The next example shows the code to use for zero page indexing.

```
0800-       LDY #$00
0802-       LDA ($3A),Y
0804-       INY
  .
  .
  .
```

Note that the indexing instruction implies a two-address indirect location; $0300-$0301 and $3A–$3B.

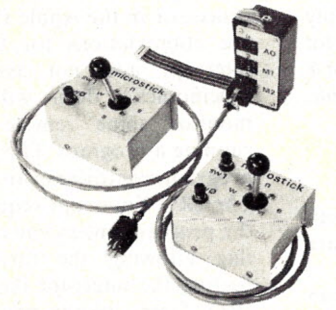
Another application of indirect addressing might be a block memory move. A typical example in the Apple II is the memory move command for the monitor. If you examine the code for this routine you will see a useful technique variation. Rather than increment or decrement an offset value, the memory addresses are incremented. Here's a short program to list a portion of memory. The routine at $FCBA is used by the memory move routine to compare byte counts.

```
MOVMEM
LDY #$00      ; index offset value
LDA (3C),Y    ; set the byte indirect
JSR $FDDA     ; print the byte in A
LDA #$A0      ; space character
JSR $FDED     ; print the character
                in A
JSR $FCBA     ; compare byte count
BCC MOVMEM    ; not done - so back
RTS           ; done - end it
```

Indexing in this example is page zero indirect. The index offset value is not changed. But, if you examine the monitor routine at $FCBA, you will see that the memory address is incremented. When the beginning address is equal to the ending address the carry flag is set. At this point the program is ended by the RTS.

If you set up a jump to the label MOVMEM address at $03F8 then the (CTRL) Y monitor function can be used. For instance, if $0800 is the starting address then at $03F8 to $03FA store 4C 00 08. To run the program enter the starting address a period and the ending

NEW FOR THE APPLE



REALIZE THE POWER OF YOUR APPLE

Enhance graphics and games with the **MICROSTIK,** control video and other hardware with the micro-box, using CJM's Modular Microsystem.

Prices start at **$44.95**

Call or write for more information

* Ask us about **CAI** applications

VISA & MC accepted

CJM P.O. Box 2367
Reston, VA. 22090 (703) 620-2444

CJM

CIRCLE 159 ON READER SERVICE CARD

SDS

presents

A.C.E.
(Applesoft Command Editor)

A powerful co-resident enhancement to Applesoft!

Functions include:

- Easy, fast line editing
- Keyboard macros
- Abbreviated commands ("L" for LIST, etc.)
- Programming Utilities:
  - single variable cross reference
  - hex/decimal convert
  - display value of all current non-array variables in use.

Commands are executed right from the keyboard without creating or loading any external files. A.C.E. is a must for efficient program development.

48 K Applesoft ROM required — $29.95
(Diskette only — please specify DOS 3.2 or 3.3)

Available from your local computer store
or

southwestern data systems
P. O. Box 582-A
Santee, CA 92071
(714) 562-3670

CIRCLE 184 ON READER SERVICE CARD

Find Your Way Around The New Apple® DOS With The Dakin5® Programming Aids 3.3®

Dakin5 Corporation, a Colorado software house, is making available to the public 12 utility programs on one 16 sector diskette, utilizing the new Apple DOS 3.3, which provides 23% more storage.

All of the Dakin5 Programming Aids 3.3 programs are also compatible with the Corvus Disk Drive system.

Features

- Remove REM statements, unreferenced (dead) code, and compress code to increase program speed and save memory and disk space.
- Copy any file or program from one diskette to another. Only the name is needed.
- Print or display a line cross reference and variable name cross reference.
- Print or display all or selected records from a text file.
- Display any sector of a given file or program, and then update any data within that sector, or specify the sector you wish to update, such as directory sectors and sectors occupied by DOS.
- Create, print and modify your own text and Exec files.
- Perform 20-digit arithmetic.
- Copy a diskette without DOS; initialize without DOS; verify source diskette; verify copied data is the same as the original.
- Use a powerful data entry routine that handles both string and numeric data.

Plus Many More Utility Programs for Sophisticated Programmers

Many of these utility programs have been developed and tested for in-house use while producing The Controller™ business package for Apple Computer Inc.

Each programming aids package includes a program diskette and very complete documentation, all attractively packaged in a padded, blue print vinyl 3-hole notebook with silver lettering. An identifying tab separates each program for convenient reference.

See your Apple dealer or contact Dakin5 Corporation, P.O. Box 21187, Denver, Colo. 80221. Telephone: 800-525-0463. VISA or MC welcome.

Apple is a registered trademark of Apple Computer Inc. The Controller is a registered trademark of Dakin5 Corporation.

DAKIN5 CORPORATION

CIRCLE 122 ON READER SERVICE CARD

GALAXY SPACE WAR I

Galaxy Space War 1 (WAR1) is a game of strategy in which the player has complete control of his space fleet's tactical maneuvers. Each fleet battles its way toward the opponents galaxy in an attempt to destroy it and win the war. WAR1 simulates the actual environment encountered in a space war between two galaxies. Optimum use is made of Apple's high resolution graphics (HIRES) and colors in displaying the twinkling stars universe, the colored ships of each fleet, long range sensors colored illuminations, and the alternating blinking colors used in battles between ships. Complementing HIRES are the sounds of war produced by Apple's speaker.

WAR1 is played between Apple and a player or between two players. You may play with total knowledge of each others fleet or only ships sensor knowledge of the opponents fleet. Each player builds his starting fleet and adds to it during the game. This building process consists of creating the size and shape of each ship, positioning it, and then allocating the total amount of energy for each ship.

During a player's turn he may dynamcially allocate his ships total energy between his screen/detection and attack/move partitions. The percentage of the total energy allocated to each partition determines its characteristics. The screen/detection partition determines how much energy is in a ship's screens and the detection sector range of its short range sensors. The attack/move determines the amount of energy the ship can attack with, its attack sector range, and the number of sectors it can move in normal or hyperspace.

When an enemy ship is detected by short range sensors, it is displayed on the universe and a text enemy report appears. The report identifies the ship, its position, amount of energy in its screens, probable attack and total energy, a calculated detection/attack/move range, and size of the ship. Also shown is the number of days since you last knew these parameters about the ship. When a ship's long range sensor probes indicate the existence of an enemy presence at a sector in space, this sector is illuminated on the universe.

An enemy ship is attacked and destroyed with attack energy. If your attack energy breaks through his screens, then his attack energy is reduced by two units of energy for every unit you attack with. A text battle report is output after each attack. The program maintains your ship's data and the latest known data about each enemy ship. You may show either data in text reports or display the last known enemy positions on the universe. You can also get battle predictions between opposing ships. The text output calculates the amount of energy required to destroy each ship for different energy allocations.

APPLE® II, 48K, APPLESOFT ROM CARD, DISK II DOS 3.2
WAR1 DISK & MANUAL ...$39.95
(CA residents add 6% sales tax)
Write or call for more information

GALAXY
DEPT. CC4
P.O. BOX 22072
SAN DIEGO, CA 92122
(714) 452-1072

Apple, cont'd. . .

address; like this —

MEMSTART.MEMEND.

Now press CTRL and Y. When you do the control Y, the monitor will jump to the program address stored at location #03F8 and the memory contents from MEMSTART TO MEMEND will be printed on the screen. Again, if you read through the program at $FCBA in the listing of the monitor, you will find that the indirect memory location at $3C - $3B is incremented. The table being indexed is the range of memory you specify by MEMSTART.MEMEND. (Note: *Indexing Principals* is rewritten from articles I wrote originally published in the Southeastern Software Newsletter.)

Nibble

Here's an Apple II information source that has something for most every Apple II owner. *Nibble* is devoted entirely to the Apple II. I get several other Apple-only publications. Usually, they are specialized or include only utility type routines. *Nibble* has a nicely balanced mixture of recreational diversions, personal and business applications, and useful hardware articles. Magazines such as *Creative Computing* will provide you with a wide-ranging perspective of computer knowledge and applications. You will need this broad scope of computer knowledge to help you develop insight and creativity. Information sources like *Nibble* help you apply your creativity, and provide you with a lot of low cost software. A subscription of 8 issues for $15.00 is available from *Nibble*, Box 325, Lincoln, MA 01773.

Head Cleaners

3M has developed a product to clean disk drive heads. Included in the kit are two special "diskettes" and a bottle of cleaning fluid. The kit allows you to clean the heads without opening the case. High-use systems should use the kit about once a month. Most of us only need to use it every 6 months or so (or if "funny" things happen to stored programs). Note that some diskettes cause more head contamination than others; especially the bargain types. In this case, you must use the kit more frequently. Order model 7440, the 5 inch size, for your Apple drives. You can get the kit from most 3M distributors or from Data Recording Products Division/3M, St. Paul, MN 55101. The cost is $30.00.

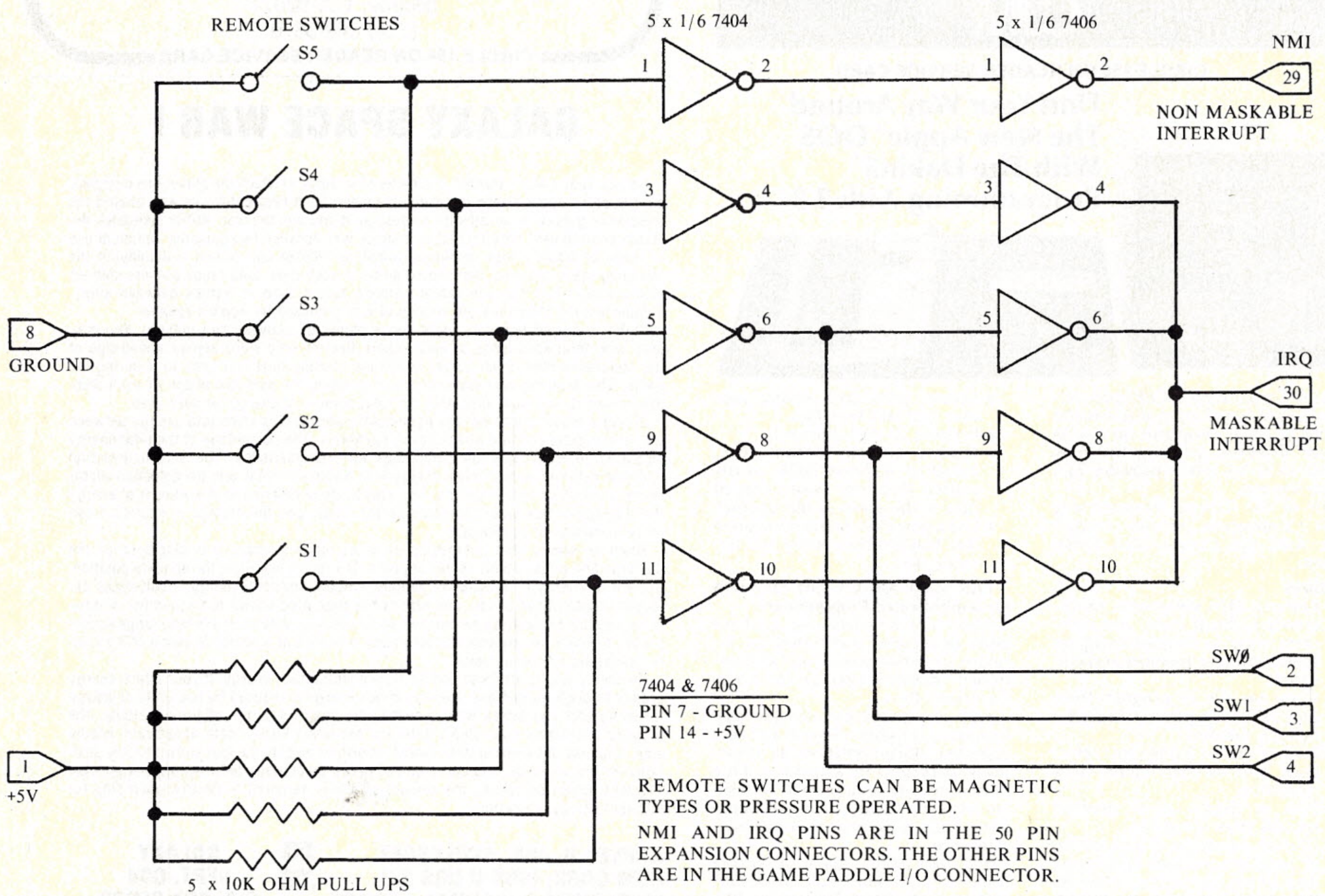
We Now Interrupt . . .

There are two types of interrupt capability in the Apple II. One is called a Non Maskable Interrupt (NMI) and the other is a Maskable Interrupt (IRQ). The interrupts are connected to the 6502 microprocessor in the Apple II. (NMI and IRQ are abbreviations for the name of the interrupt and not assembly language mnemonics.) Both will allow you to monitor some remote function while running a program. The NMI will halt the program regardless of any other condition. The IRQ (Interrupt request) will not halt the program unless you clear the interrupt flag allowing the interrupt to occur. Setting the interrupt flag will prevent an IRQ from taking control. From this discussion you can see that NMI is the highest priority interrupt.

Interrupt Access

Both of the interrupts are available from the expansion connectors; pins 29 and 30 for NMI and IRQ respectively. Usual access is made through the edge connector of a circuit board made to plug into the expansion connector. One such board is the Apple prototyping board. This is an expensive way to connect to just two

Figure 3. Interrupt Control Circuit.



USE A CONTROL PROGRAM LIKE LISTING 1 TO SENSE THE IRQ INPUTS. A SIMILAR PROGRAM AT A DIFFERENT MEMORY LOCATION CAN BE USED FOR NMI. THE MULTIPLE INTERRUPT INPUT COULD ALSO BE USED FOR NMI. JUST CONNECT THE 4-SWITCH CIRCUIT TO PIN 29 INSTEAD OF PIN 30. THE PROGRAM IN LISTING 1 WOULD BE THE SAME. THE S5 CIRCUIT CAN BE LEFT OFF IF ONLY ONE INTERRUPT TYPE IS USED.

For your Apple II....

MUSIC & GRAPHICS

ALF Music Synthesizer

The ALF music synthesizer has three voices on each board which are easily programmed using the Entry program provided. The envelope shape of each voice (or even each note) may be controlled individually thus allowing the synthesis of practically any instrument such as a violin, trumpet, piano, harp or bells. Instrumentation and dynamics may be varied while a song is playing by changing the attack, sustain, release, decay, gap and volume of the notes.

Playback of music is accompanied by a spectacular color display showing a stylized "piano keyboard" for each part with the colors of the notes varying in proportion to their loudness and waveform.

Ease of Music Entry

Music is entered directly using the high-resolution graphics entry program. One paddle is used to select menu items such as note duration, accidentals, dotted notes, triplets, tied notes, etc. while the other paddle moves a note cursor up and down the staff over a 4-octave range. The transpose command extends the range to eight octaves. This form of music entry is considerably faster and more accurate than cryptic note code schemes (like QFS3) found with other synthesizers.

MUSIC ENTRY SCREEN

The board plugs into any Apple II or Apple II Plus. Two or three boards are required for stereo. Requires a 16K Apple system and external amplifier and speakers.

***ALF Apple Music II (AM-II) Synthesizer**

The AM-II is a new, low cost digital music synthesizer for the Apple II computer. It features 9 voices on a single music card.

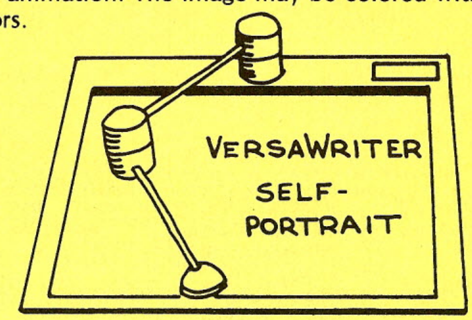
The software ENTRY and PLAY systems are the same as on the ALF Apple Music Synthesizer (AMS). The two principle differences between the new AM-II and the original Apple Music Synthesizer are in pitch range, volume range, and parts per board.

The new AM-II has a range of six octaves. The dynamic range is 28 db. (The original AMS has a range of 8 octaves a dynamic range of 78 db and 3 parts per board.)

VersaWriter

VersaWriter is a drawing tablet for the creation of full-color, high resolution graphic images on the Apple. Images may be drawn freehand or traced from existing images (cartoons, photos, drawings, etc.) using the simple pivoted two-arm pantograph with magnifying crosshairs.

After an image is drawn, it may be rotated, shrunk, or enlarged. It may be moved across the screen and alternated with other images thus providing high-resolution animation. The image may be colored with varied colors.



Animate other Programs

Graphical images made with VersaWriter and stored on tape or disk may be called from other programs or even imbedded in them. With VersaWriter, you don't have to worry about assembly code, counting pixels or other cumbersome hi-res graphics entry and retrieval techniques.

VersaWriter graphics can be used in all types of programs—games, statistics, engineering, artistic, and educational. Your only limit is your own imagination.

Two Disks of Software

Disk 1 contains the basic plotting, scaling, movement, rotation, color, transfer and recall software. This disk also includes routines which create "shape tables" from your figures to be used in other programs. Disk 2 contains applications software. One program adds five sizes of upper and lower case text to drawings, another adds standard electronic and digital symbols, while a third calculates distances and areas.

VersaWriter requires a 32 or 48K disk system, Applesoft in ROM or an Apple II Plus.

VersaWriter	$252.00
ALF Music Synthesizer	$268.00
AM-II Synthesizer	$198.00

Prices postpaid in USA. NJ residents add 5% sales tax.

To order VersaWriter or the ALF Synthesizer, send your name and address along with a check or chargecard number and expiration date. Visa, MasterCard and American Express are welcome. Units are in stock and orders will be shipped as soon as your check clears or your credit is verified.

Dealer inquiries invited.

Peripherals Plus

119 Maple Ave., Morristown, NJ 07960 (201) 538-3385

CIRCLE 252 ON READER SERVICE CARD

Apple, cont'd. . .

pins. One idea would be to find a scrap circuit board with the proper edge connector. Cut the connector off and solder wires to the two interrupt pins. Connect these wires to a 16 pin component header plugged into the game paddle I/O socket. Use the two pins which are not connected to other circuits (9 and 16). Some of the other pins will be used for other interrupt monitoring connections.

Another possibility would be to remove the main circuit board and jumper the interrupt lines to the unused GP I/O socket pins. If you or someone you know has experience making jumpers on circuit boards, this is the best way. You could mess up the board so don't try it unless you know what you're doing. Doing this will void warranties, too. The connections to the GP I/O of the NMI and IRQ signal lines make it possible to make other simple connections for monitoring more than two devices. We'll get to that shortly.

Signal Levels

Both the interrupt lines have pull-up resistors. This means that an active low signal is required to cause an interrupt to occur. The NMI is edge sensitive. When an input of the proper direction occurs, the interrupt occurs. The duration of the input signal is not important. Only the leading edge of the signal is sensed. However, no other input to NMI can occur until the current interrupting signal returns from low to high. The IRQ on the other hand is level sensitive. A signal change to active low for some period of time is required to make the interrupt occur. The minimum amount of time is the length of the longest instruction cycle. This is because the interrupt does not happen until the current cycle is completed. Again, the signal must go away before any other maskable interrupts can occur. A useful reference on the discussion of 6502 interrupts can be found in *Micro* magazine for July 1980, page 47.

An Interrupt Program

Listing 1 is an assembly language program for polling a series of switches used in the circuit of Figure 3. This circuit allows monitoring of the non maskable interrupt as the priority input. It also provides for polling of 4 inputs to the maskable interrupt. Since we are using the three switch input bits in the GP I/O connector, we can have 4 input circuits, the assumption being that we can test three inputs directly and default to the fourth. The flow chart in Figure 4 illustrates the assumption. Before we continue with a discussion of the interrupt program, a short digression is needed to establish the maskable interrupt initialization (if you are to include this capability in a Basic program).

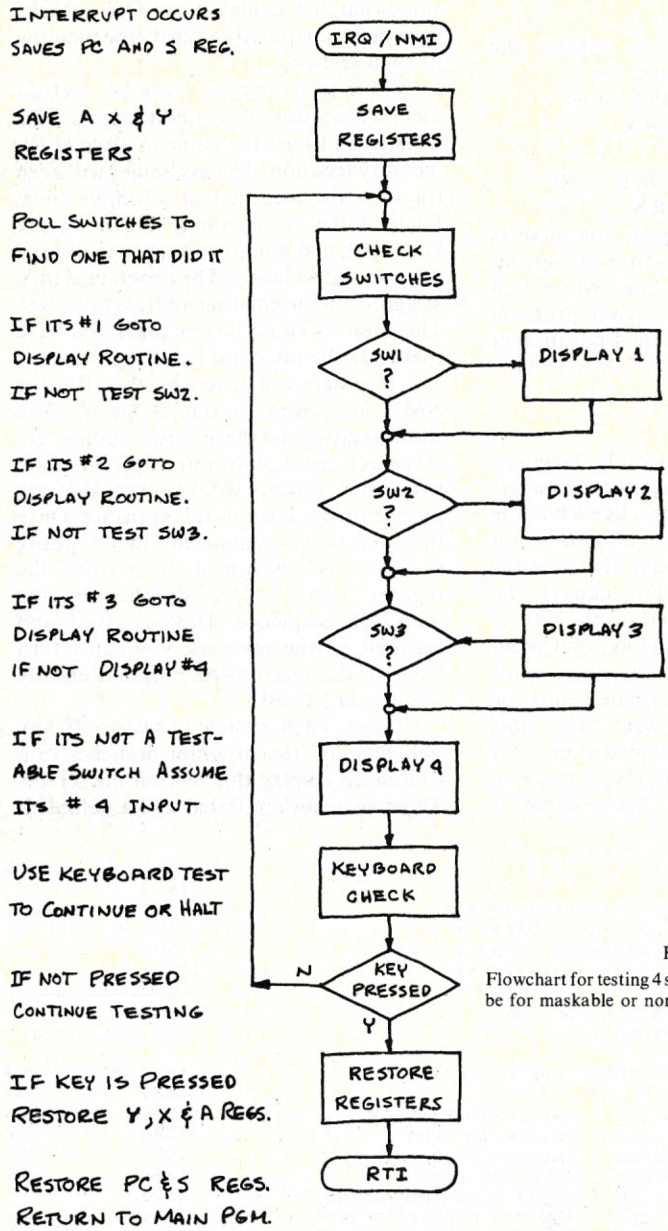


Figure 4.

Flowchart for testing 4 sources of interrupt. Polling can be for maskable or non maskable interrupts.

Initializing Interrupts

The Apple monitor is the first place where interrupts are processed. A jump indirect through the address stored at locations \$03FE and \$03FF occurs as the result of an IRQ input. The address of the interrupt handling routine would be stored at these addresses. Addresses \$03FB to \$03FD are used to contain a jump to the address of the program used to handle non maskable interrupts. The addresses can be the same or different depending on your needs for interrupt handling. Our example program in Listing 1 is for IRQ input. The handling requirements are similar so the program will serve as an illustration for IRQ and NMI application.

In addition to initializing the vector addresses to the handling programs, the IRQ flag must be set. The mnemonic instructions used to do this are CLI (58 hex, 88 dec) to allow the IRQ to interrupt the program, and SEI (78 hex, 120 dec) to prevent interrupts from the IRQ input. As you might guess, the mnemonics stand for clear interrupt and set interrupt. In a Basic program the initialization can be handled by a series of POKEs to memory. To poke the interrupt routine address use

```
xxxx POKE 1022,00
xxxx POKE 1023,03
```

to put the IRQ routine at address \$0300. And, use

```
xxxx POKE 1019,76
xxxx POKE 1020,00
xxxx POKE 1021,03
```

to put the NMI routine at address \$0300. Note that 76 is the decimal value of the op code \$4C (mnemonic JMP). Additionally, provide a way to set or clear the interrupt flag as desired when you start the program. You can use pokes to do this too.

Apple, cont'd. . .

Following an input line asking the program user which choice, use

xxxx POKE 10,88:POKE 11,60
xxxx CALL 10:RETURN

to allow interrupts and,

xxxx POKE 10,120:POKE 11,60
xxxxCALL 10:RETURN

to prevent interrupts. Again, use these as part of your initialization program. Addresses 10 and 11 ($A and $B) are used for the USR function. If your program includes the USR function find another pair of addresses to use.

The Program

Listing 1 is an assembly language program to poll several inputs to the single IRQ line. This program checks each of the three switch inputs. If one of them is not on, the assumption is made that it is the fourth. Whichever input causes the interrupt will be displayed on the screen. Should any other switch close, that input will be displayed too. All inputs must return to the normal open state before the cycle can be started over. Since this program and circuit are intended only for demonstration, no attempt was made to provide automatic reset. The program is somewhat self explanatory. Only a brief comment is required to clarify the function of each section.

The first section equates actual memory locations to symbolic names. This allows you to use the name in place of the memory location. The assembler will keep track of the locations at assembly time. Each of the switch input addresses, the keyboard, and monitor routines to be used are equated to labels. The program is then assigned the originating address of $0300. This address could be any place you have space for the interrupt handling routines.

Registers not saved by the IRQ (or NMI) are saved in this sequence. The status register and the program counter are saved as a result of the interrupt. To insure that other registers will be saved, they are pushed on the stack by this sequence. Once the routine is completed, the opposite sequence is performed to restore the registers. This is accomplished by the RESTOR sequence. Having saved and restored all the registers, you can return back to the interrupted program exactly where you left off.

Next, each switch is tested. If any switch is on, the program branches to a routine to display this fact on the screen. Three switches are tested and a default is made to the fourth interrupting device. The assumption is made that this routine is running because an interrupt occurred. Therefore, if it's not one of the testable switches, it must be the one left.

Having tested and displayed the switch indicating the interrupt source, the program returns and repeats the test of the inputs. But, not before the keyboard input is tested for a pressed key. If no key has been pressed, the routine continues. Should any other input switches close, they will be displayed too. Response to the reason for the interrupt can be made at this time. If a key is pressed, the program passes to the register restore section. The key-testing routine permits you to allow the interrupt condition to continue until you have made whatever action is necessary.

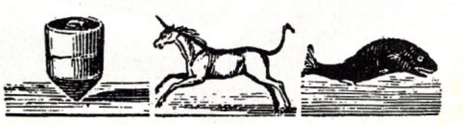
Following the restoring section is the common routine for displaying the response. Depending on the activated circuit, a register is loaded with the switch number. This number is then used by the following routine to display a message and indicate the switch number. The alarm message uses the absolute indexing method mentioned at the beginning of this column. The message is contained in an ASCII string at the end of the program. □

Listing 1A.

```
LIS 1000,1500

1000 *****************************
1010 * INTERRUPT POLLING ROUTINE *
1020 * BY: CHUCK CARPENTER  7/80 *
1030 *****************************
1040
1050 *  USES THE SYNTAX OF THE
1060 *  S-C ASSEMBLER II
1070
1080 *  SYMBOLIC ADDRESS ASSIGNMENTS
1090
1100 SW1     .EQ $C061   SWITCH-IN PIN2 GP I/O
1110 SW2     .EQ $C062   SWITCH-IN PIN3
1120 SW3     .EQ $C063   SWITCH-IN PIN4
1130 KEY     .EQ $C000   KEYBOARD DATA
1140 STROBE  .EQ $C010   CLEAR KEYBOARD STROBE
1150 CHROUT  .EQ $FDED   MONITOR CHARACTER OUT
1160
1170         .OR $0300   PAGE 3 ORIGIN
1180
1190 *  SAVE THE REGISTERS
1200
1210 SAVE    PHA         SAVE ACCUMULATOR ON STACK
1220         TXA         PUT X IN A
1230         PHA         SAVE X ON STACK
1240         TYA         PUT Y IN A
1250         PHA         SAVE Y ON STACK
1260
1270 *  CHECK THE SWITCHES AND DISPLAY
1280 *  THE ONES THAT ARE ON
1290
1300 CHK1    LDA SW1     CHECK SWITCH 1
1310         BPL CHK2    NOT ON-GOTO SW2
1320         JSR DISP1   ON-GOTO DISPLAY 1
1330 CHK2    LDA SW2     CHECK SWITCH 2
1340         BPL CHK3    NOT ON-GOTO SW3
1350         JSR DISP2   ON-GOTO DISPLAY 2
1360 CHK3    LDA SW3     CHECK SWITCH 3
1370         BPL CHK4    NOT ON - MUST BE 4
1380         JSR DISP3   ON-GOTO DISPLAY 3
1390 CHK4    JSR DISP4   DISPLAY BY DEFAULT
1400
1410 *  PRESS A KEY TO ESCAPE
1420 *  FROM POLLING ROUTINE
1430
1440 KYBD    LDA KEY     CHECK FOR PRESSED KEY
1450         BPL CHK1    NOT PRESSED-BACK TO SW TEST
1460         STA STROBE  PRESSED-CLEAR STROBE
1470
1480 *  RESTORE THE REGISTERS THEN
1490 *  BACK TO MAIN PROGRAM
1500
:
```

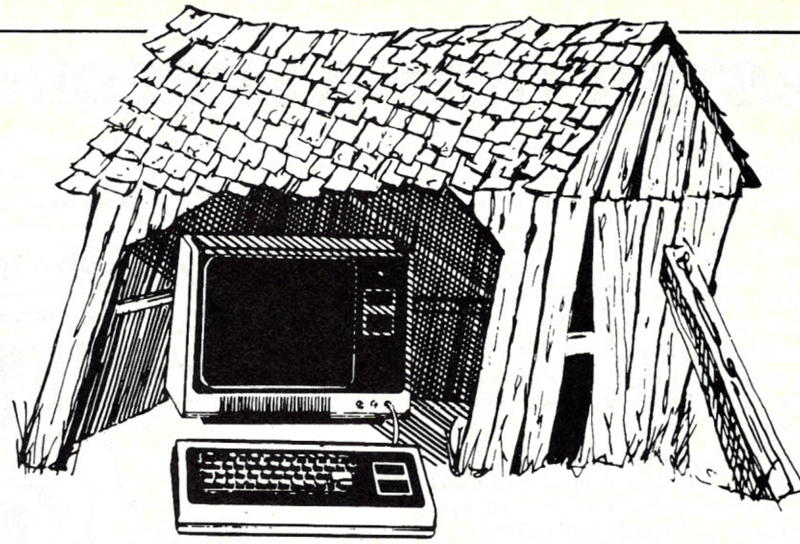
Listing 1B.

```
LIS 1510,1870

1510 RESTOR PLA          GET Y BACK TO A
1520        TAY          PUT A IN Y
1530        PLA          GET X BACK TO A
1540        TAX          PUT A IN X
1550        PLA          GET A BVACK
1560        RTI          GOTO MAIN PROGRAM
1570
1580 *  PUT THE SWITCH NUMBER INTO
1590 *  TEMPORARY STORAGE
1600
1610 DISP1  LDA #$31     SWITCH 1 ON
1620        STA $FA      STORE IN IN SCRATCH LCTN
1630        JMP CRT     JUMP TO DISPLAY RTNE
1640 DISP2  LDA #$32     SWITCH 2 ON
1650        STA $FA      STORE IT
1660        JMP CRT      JUMP TO DISPLAY
1670 DISP3  LDA #$33     SWITCH 3 ON
1680        STA $FA      STORE IT
1690        JMP CRT      JUMP TO DISPLAY
1700 DISP4  LDA #$34     SWITCH 4 ON
1710        STA $FA      STORE IT
1720
1730 *  PRINT THE ALARM MESSAGE
1740
1750 CRT    LDY #$0F     LOAD INDEX DISPLACEMENT
1760 CRT1   LDA TABLE,Y  PRINT THE MESSAGE
1770        ORA #$80     SET HI BIT - NORMAL VIDEO
1780        JSR CHROUT   PRINT THE CHARACTER IN A
1790        DEY          NEXT CHARACTER
1800        BPL CRT1     BACK FOR MORE TABLE
1810        LDA $FA      LOAD THE ON-SWITCH #
1820        JSR CHROUT   PRINT IT
1830        LDA #$8D     LOAD A CARRIAGE RETURN
1840        JSR CHROUT   PRINT IT
1850        RTS          BACK TO INTERRUPT ROUTINE
1860 TABLE  .AS " HCTIWS TA MRALA"
1870        .EN
:
```



TRS-80 Strings

Stephen B. Gray



For this two-dozenth column, we look at Radio Shack's three new TRS-80 computers, eight game cassettes from Programma International, and a subroutine that adds fireworks to any program.

Now We Are Five

If you're interested in buying a Radio Shack computer, you now have five TRS-80 models to choose from, because three new ones were introduced on July 31:

- TRS-80 Model III, almost a Model I with room for dual integral 5¼-inch disk drives, starting at $699, which places it between the Models I and II.
- TRS-80 Color Computer, beginning at $399, and using plug-in ROM "Program Paks" or cassette or disk for input, along with joysticks.
- TRS-80 Pocket Computer, 7 inches long, weighing 6 ounces, fits in your hand or pocket, $249. The size of an electronic translator, it connects via a $40 interface to a 7½-inch-long $79.95 cassette recorder and most other compatible recorders.

TRS-80 Model III

The Model III combines the integral-disk-drive features of the Model II with the low-end price of the Model I. It can be programmed either in Level I Basic, which is Palo Alto Tiny Basic, or in Model III Basic, which is Radio Shack's revision of Microsoft Basic, with some of the features that were removed to make the Model I Level II's Basic fit into 12K of ROM.

Model III is housed in a single cabinet, with a keyboard, numeric keypad, 12-inch video monitor, and space for two built-in double-density disk drives. The basic Model III comes without drives; what looks like drives in the photo are plastic covers molded to look like the front end of disk drives, but labelled

REPLACE WITH
RADIO SHACK DISK DRIVE

The disk drives are the first products of the joint venture between Tandy Corp. and Datapoint, called Texas Peripherals.

Three versions of the Model III will be manufactured: Level I Basic system with 4K RAM, $699; Model III Basic system with 16K RAM, $999; Desktop Business Computer with 32K RAM, two 40-track disk drives with 313K of user storage, and a built-in RS-232 serial interface, $2,495.

Model III has the same display of 16 lines of up to 64 characters each as the Model I, and is also a Z80A machine. In the models at and above $999, you get upper and lower case.

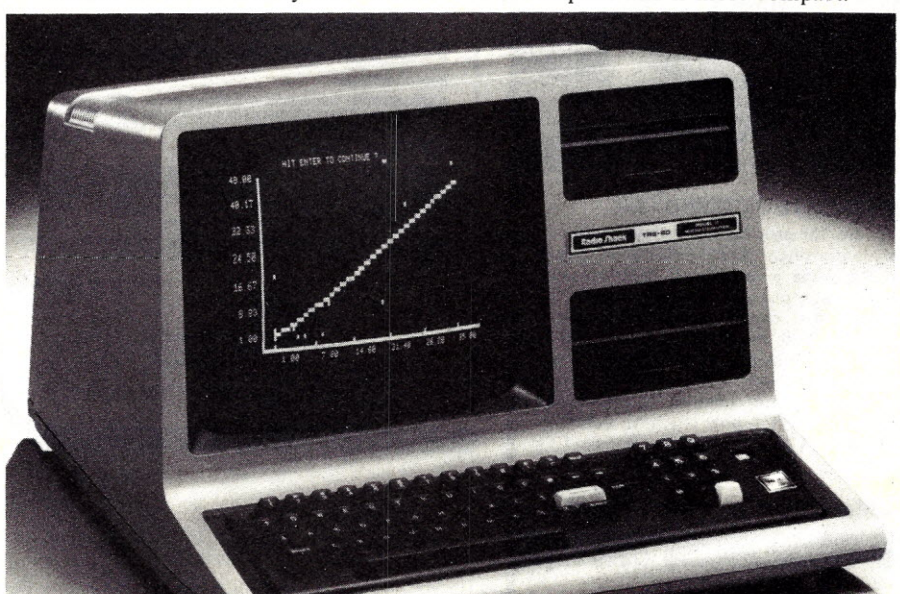
The Desktop Business Computer can be expanded to 48K RAM and two additional disk drives. It will read Model I disks. Model III Basic is compatible with most Model I programs. Radio Shack has prepared many programs for Model III, including Scripsit.

Model III has a totally new circuit board and a redesigned architecture. There is some memory-mapping, but it uses input/output through ports instead of memory addresses.

The daisywheel printer used in the $4,500 Model III word-processing configuration is a Japanese-built 43-cps Ricoh RP 1600, priced separately at $1,960, about $1,000 less than comparable daisywheel printers by Qume, Diablo and NEC.

Also announced were a bidirectional Line Printer VI that prints upper and lower case characters in four type sizes, plus graphics characters and special symbols, operating at 100 cps and priced at $1,200; and a $1,500 plotter/printer programmable in Basic.

Model III should provide a great deal of competition for the Apple III. Although it displays fewer rows and columns on the CRT, and has less graphics capability, it is lower in price and is more compact.



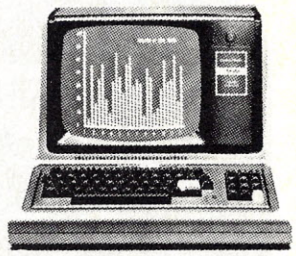
MODEL II

26-4002
64K 1 Drive
$3499.00

AUTHORIZED
TRS-80® DEALER A301

COMPUTER SPECIALISTS

MODEL I



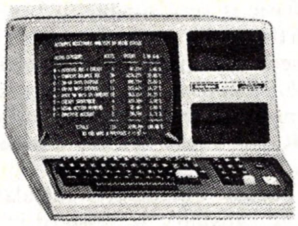
26-1054
4K Level II
$552.00

26-1056 16K Level II System with Keypad. $670.00

26-1145 RS-232 Board	84.00
26-1140 "O" K Interface	249.00
26-1141 "16" K Interface	365.00
26-1142 "32" K Interface	476.00
26-1160 Mini Disk - Drive O	424.00
26-1161 Mini Disk - Additional	424.00
26-1154 Lineprinter II	720.00
26-1156 Lineprinter III	1799.00
26-1180 Voice Synthesiser	339.00
26-1181 VOXBOX	145.00
26-1104 Factory Upper/Lower Case Modifcation Installed	70.00
26-1506 Scripsit - Tape	60.00
26-1563 Scripsit - Disk	85.00

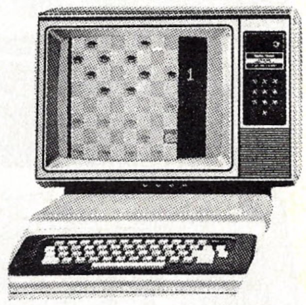
NOTE: Call for availability of VIDEO TEX, Model III, Color, and other new products.

MODEL III



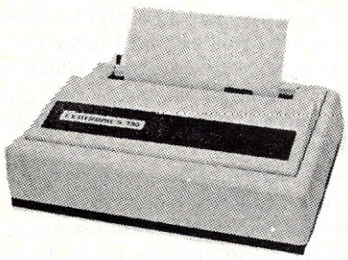
26-1061 4K I	$630.00
26-1062 16K III	900.00
26-1063 32K III 2-Drives, RS232	2246.00

COLOR



26-3001 4K	$360.00
26-3002 16K	540.00
26-3010 Color Video	360.00
26-1206 Recorder	54.00
26-3008 Joysticks	22.50

ALL OTHER R.S. SOFTWARE FURNITURE, STANDS, CABLES AND ACCESSORIES DEDUCT 10% FROM CATALOG PRICE



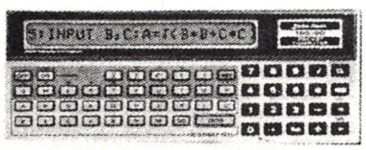
CENTRONICS

Fast 100 CPS Centronics 730 Printer.................. $675.00
Text Quality Centronics 737 Printer.................. $850.00

Model II Cobol Compiler $360.00
Cobol Run Time Package $36.00

Novation Cat Modem..$149.00
CCA Data Management System..................72.00
Adventure Games Games 1-9 each.........14.00

Pocket Computer



26-3501 1.9K P.C.	$225.00
26-3503 Cassette I/F	45.00
14-812 Recorder	72.00

Acorn Software Products, Inc.

GAMES:

Alien Invasion	$9.00
Stock Market	9.00
Star Trek	9.00
Block 'Em	9.00
Ting-Tong	9.00

UTILITIES:

System Savers	14.00

EDUCATION:

Language Teacher	18.00

FREE: PRICE LIST UPON REQUEST

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC.

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No.

Full Factory Warranty on All Items Sold.

Largest Inventory In the S.E. U.S.A.

*TRS-80 is a registered trademark of the Tandy Corp.

CIRCLE 163 ON READER SERVICE CARD

TRS-80 Color Computer

The Color Computer is Radio Shack's first TRS-80 to use a processor other than the Z80A, and is the first TRS-80 to use plug-in ROM software cartridges.

At $399, the Color Computer replaces the Model I at the low end of the TRS-80 line. You hook it up to your color TV set via an RF modulator, or buy Radio Shack's new $399 color TV set. A 16K Extended Basic Color Computer is $599.

The first ROM Program Paks will cost from $29.95 to $39.95, and will be initially available for education, entertainment and home applications.

The Color Basic is a version of Level II Basic without double precision.

The Color Computer uses Microsoft 6800 Basic converted to 6809 Basic; the processor is a Motorola 6809E. Rumor has it that the Color Computer was originally designed to use the Z80A, but then Motorola came up with the 6809, a new video-display generator chip, and "a considerable amount of design assistance," according to one source.

The Color Computer actually contains two processors, one for arithmetic and the other for Basic interpreter and key-in.

The Color Computer provides eight colors, and five different resolutions, from 32x64 to 296x256. Some of the three intermediate resolutions require 16K and Extended Basic.

As I understand it, there are no special graphics characters such as the graphics block and the 63 graphics characters provided by the TRS-80 Model I, Level II.

TRS-80 Pocket Computer

The Pocket Computer features a 24-character LCD display with English-language prompting and Basic programming. It includes 1.9K bytes of RAM that hold the information for the 300-hour life of the internal batteries. It's supposed to be able to perform "almost any of the smaller jobs the TRS-80 Model I can do."

Pre-programmed cassette tapes for the Pocket Computer include real estate ($24.95), civil engineering ($24.95), personal finances ($19.95), aviation ($24.95), math drill ($14.95), and a games pack ($14.95). Cassettes may also be used to store programs and data.

The TRS-80 Pocket Computer, which is made for Radio Shack by Sharp Electronics Corp., can also be used as a calculator. Numbers can be edited, stored, reviewed and placed in equations with up to 15 levels of parentheses.

Radio Shack has a one-year exclusive agreement with Sharp to market the Pocket Computer. A printer is scheduled for later.

The Basic is very similar to Level I Basic, with the addition of string functions, 15 arithmetic functions, and transcendentals. Sharp copied Level I when they originally designed the computer, according to a Radio Shack source, who says "Level I is very popular in Japan."

The Pocket Computer's calculator ancestry and capability is reflected in its 1,424-step memory, a memory with 26 "data elements," and a 48-step reservable memory for storing frequently-used functions.

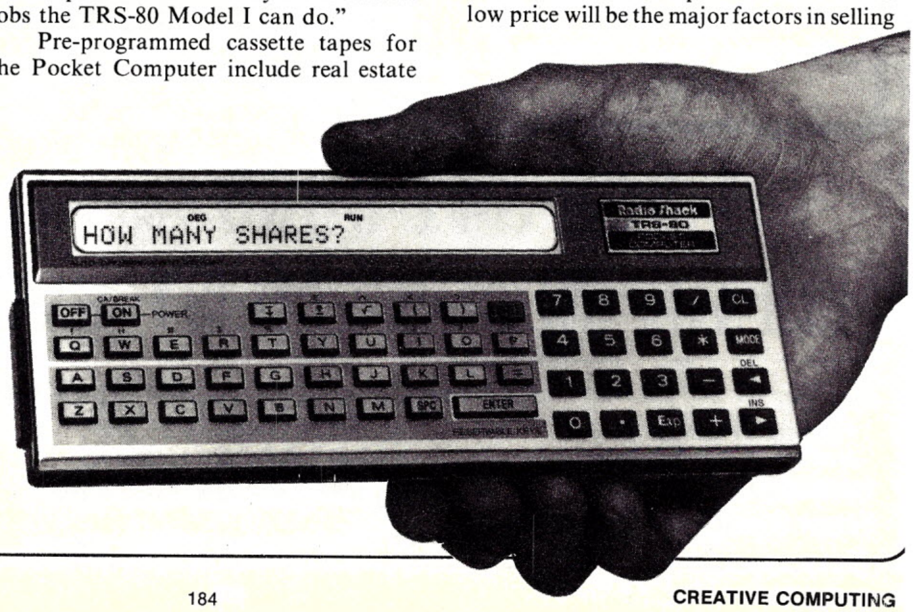
The alphabetic keys are arranged in standard typewriter QWERTY format. The keys are about as close together as the keys of the first PET computer, which were actually calculator keypads. So unless you've got tiny fingers and/or unusual dexterity, you may not be able to enter programs with more than two fingers, in the classic "hunt and peck" fashion.

An automatic power-off feature saves battery life if no entry is made within 7 minutes.

Although some critics, such as Adam Osborne, have dismissed the Pocket Computer as a "stupid product" and "junk," Radio Shack will probably sell many thousands, perhaps even more than the Model III and the Color Computer.

PC and CC

The Pocket Computer's small size and low price will be the major factors in selling



When DecisionMaster™ speaks everybody listens.

Let's face it. We all have to make decisions. Decisions that can change our lives. Decisions that can make us happy or unhappy. Decisions that could win us fame or fortune. Now, DecisionMaster can help you make the best decisions of your life.

Use Bayesian theory to peer into the future...even if you've never heard of the Bayes' Rule. Do a complete weighted factor analysis...without knowing what one is. Use discounted cash flow to compare investment alternatives without bothering with present value tables. These and other sophisticated theories that were once the exclusive domain of professors and top business executives are *built into* DecisionMaster's algorithms... so you can use them at the touch of a key!

DecisionMaster is easy to use. It features:

- A fully documented manual developed by an authority in the field.
- A unique program-controlled cross reference system.
- A powerful formatted-screen data entry system.

You'll use DecisionMaster in hundreds of routine decisions, as well as more important ones such as • Buying a house • Changing jobs • Selecting investment • Evaluating insurance policies • Expanding product lines • Leasing vs. purchasing.

If you buy only one computer program this year, make it DecisionMaster. And when it speaks, *listen*.

CIRCLE 200 ON READER SERVICE CARD

DECISION?MASTER™

To order, see your software dealer or return this form with your check to: DecisionMaster, Dept. CC, 10428 Westpark, Houston, Texas 77042 Add $1.50 for shipping and handling

Diskette for: ☐ Apple II* (32K) $29.95 ☐ TRS-80** (Level II-32K) $29.95 ☐ TRS-80** (Model II) $39.95

Name

Address

City State Zip

CHARGE by phone toll-free: **1-800-231-5768 Ext. 306**

(In Texas: 1-800-392-2348) or return this form:

☐ VISA ☐ MasterCard Bank Code

Account No.

Signature

Expiration Date

*Apple II is a registered trademark of Apple Computers, Inc.
**TRS-80 is a trademark of the Radio Shack Division of Tandy Corp.

TRS-80, cont'd . . .

it to the many who will prefer portability to all else. It will probably come to be known as Radio Shack's PC, and the Color Computer will be called the CC, and jokes will be made about both nicknames on the order of the "Trash-80" epithet. Meanwhile, both will sell better than any other company's computers, probably outselling both the Apple II and PET.

Programma Games

A variety of games and utility programs for the TRS-80 (and PET and Apple) is available at your local computer store, or from Programma International Inc. (3400 Wilshire Blvd., Los Angeles, CA 90010).

Of the eight Programma Level II 16K games I've seen, six are $6.95, and two (Pentominoes and Maze-80) are $9.95. Actually, only six of the eight are really games; Phonegrams is more of a demo program, and The "I Ching" Thing tells fortunes.

Of the eight, I found Tank to be a very good two-player game, Maze-80, Fifteen Numbers, and Pentominoes to be fine (and somewhat addictive) for the solo player, Pachinko and Tribble Trap to involve as much (if not more) luck as skill, Phonegram to be prosaic except for the Digit-Rotate feature, and The "I Ching" Thing to be much better for parties than for solitary use.

Tank

The full name of this game is Tank Search and Capture. The object is to collect as many points as possible before time runs out.

Numbers appear at random locations on the field, and the first player to drive his tank over the number will add that number to his score.

Complications include not being able to travel through any walls, and the numbers being harder to find among the letters that clutter up the field more and more, as the game progresses.

This is one of the better Programma games, because it's almost entirely based on skill, and after some individual practice to learn how to use the control keys with some fluency, you've got a good chance of winning.

Maze-80

"The object of Maze-80 is to find your way thru the 8 mazes . . . To move within the maze, use the arrow keys." Sounds simple.

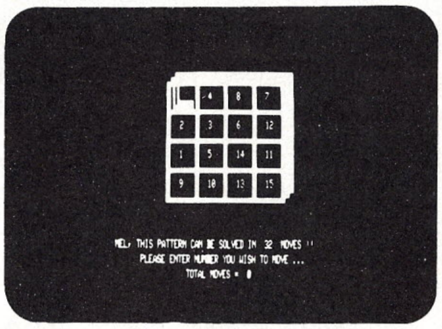
But wait — "as soon as each maze appears, the computer will start moving thru in the opposite direction. You must reach the exit on the right before the computer finds the exit on the left, or you lose!"

So as you control one blip, the computer sends its blip in your direction, trying to get out before you do. If you stop to trace out the maze backward, from exit to entrance, the computer's blip may get so far ahead, you'll never catch up.

This game can be addictive, and is not recommended for those with high blood pressure.

Fifteen Numbers

This program is called a "3-dimensional graphics version of that age-old game of Magic Squares." However, it is not magic squares, which requires adding up the numbers in the squares in various directions. Nor is it really 3-D; lines have been added in three places to make the display seem three-dimensional, but the effect isn't very good.

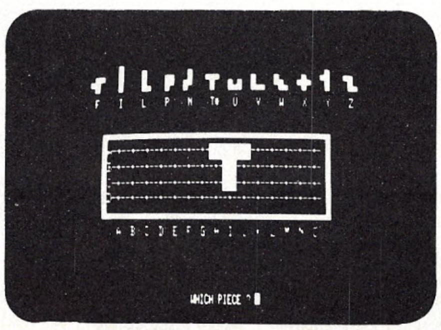


This is a computerized version of the little plastic game with 15 small numbered squares that you move around within a plastic frame, one at a time, trying to get them into different arrangements.

This is one of those programs you can run dozens of times before you catch on to the basic tricks of how to get squares into the locations where you want them. But then the program gives you another arrangement. To make things worse, the program also tells you in how many moves the pattern *can* be solved.

Pentominoes

This is a geometric puzzler. As the screen instructions put it, "A pentomino is made by joining five squares . . . There are 12 pentominoes . . . You will be trying to put all 12 pentominoes together to form a larger shape — much like doing a jigsaw puzzle. Unlike most jigsaw puzzles, pentominoes may be flipped over and used upside down."



You select one of six rectangles of varying dimensions, and then try to fill *all* the space within it with pentominoes, using each one only once, and using R to rotate a piece 90 degrees, F to flip a piece, etc.

This is a highly challenging game, and filling even the easiest of the six rectangles could take hours, even if you're quite good at geometry. This may be good training for those IQ tests involving geometric figures.

Pachinko

Pachinko is a vertical pinball machine, with seven pockets protected by slanting rows of graphics blocks. The program simulates the balls' random paths downward.

You start with 15 balls, and shoot by just pressing any key — no skill is involved. Getting a ball in a pocket wins you 15 more balls; "to break the bank, you must get 500 or more balls to your credit."

Good programming, but since no skill is involved, this may be more suitable to the very young.

Getting 500 balls isn't all that difficult, either; it took me less than five minutes to "break the bank."

Tribble Trap

The game is to capture alive a Tribble, which is a "small furry critter" that has a "mysterious healing power." It "can move in any direction, but only one sector at a time."

"To capture the elusive beast, you must shoot out the sectors surrounding him." There are further complications.

You're given a grid of 7 by 7 squares, and the "last known sector" of a Tribble. The point is to trap the Tribble, not blast him. So you try to guess the next direction he'll take. Not easy. The Tribble always seems to head for the sector you're about to blast.

There can't be a great deal of skill involved here, since the Tribble slips away on random paths. To trap one seems to require as much luck as skill, if not more.

Phonegram

For $6.95 you can find a mnemonic for any telephone number you want to remember. Providing, of course, that the number translates to something like FLYAWAY instead of GFLWYRH.

Also, as the spec sheet says, "This is a very good demonstration to friends of computing power because you show them the computer actually doing something useful for them personally."

With Phonegram, you enter the 7-digit number, and the menu asks if you want (1) single-column format, (2) three-up format, or (3) the Digit-Rotate feature.

Single-column format prints all possible combinations of the alphabetic equivalents of the telephone number entered, printing 0 or 1 if these are in the

number, "because no letters exist for these numbers on the dial."

Three-up format prints three columns across the screen, instead of one.

The Digit-Rotate feature is the best part of this program. It first converts your number to one possible set of letters. Then, by touching the keys 1 to 7, you can change any of the letters to either of the other two possible letters. This way, you can find out if there's a suitable mnemonic for the number much faster than by watching hundreds of groups of letters flash by on the screen.

This program is more for fun than serious use, because you could do it much faster (and cheaper) with pencil and paper. However, that third item on the menu is very clever.

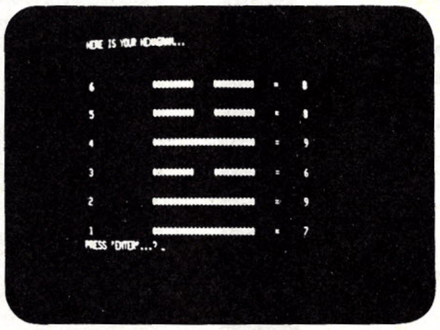
The "I Ching" Thing

This program "is probably the oldest (and most accurate) divination technique," according to the displayed instructions. "It uses 64 hexagons to form patterns that represent your fortune. Until recently, anyone consulting the 'I Ching' had to use wooden sticks and a book to read their fortune."

I Ching consists of two programs: "one to cast the hexagram, the other a data file with fortunes."

Programma presents I Ching totally straight: one display says "We will use an electronic method to reach your subconscious mind. Our goal will be to contact your higher self or super-conscious."

The program asks you to "quiet your mind," then "formulate your question," then "think about it, form an image. Then press any number on the keyboard and press ENTER."



You do this six times; that is, you press six numbers and the program then generates a hexagram, based on a random-number generator that uses your six numbers.

The programs informs you that you've cast hexagram 28, for example, then goes on to "calculate a new hexagram based on the old one."

When I entered the question "How good is this program?" and used the numbers 1 through 6, the result was hexagram 37, which was recast to form hexagram 28.

The reading for the first hexagram was "Developing. Some place." For the second, "Excess. Out of range." If that makes any sense to you, buy this program, which contains 64 of these pairs of words or phrases.

Short Program #13

Dick Spicer of Windsor, Ontario, Canada, sent this:

"Here's a subroutine for the TRS-80 that my seven-year-old likes. I use it in a mathematics and multiplication quiz program. When he gets 10 out of 10 right, the program branches to this sub. I thought some of your readers might like to add it to their game programs.

```
100 CLS
110 PRINT "YOU DESERVE"
115 FOR Q=1 TO 500: NEXT Q
117 CLS
120 PRINT CHR$(23) "FIREWORKS"
125 FOR Q=1 TO 500: NEXT Q
130 FOR L=1 TO 300
140 P=RND(1023)
150 PRINT @ P, "*        *       BANG"
155 NEXT L
160 CLS
170 RETURN
```

"Leave about 8 to 10 blank spaces between the asterisks and also before the BANG in line 150." □

Reviews

Steve Gray, et al

Computers in Mathematics: A Sourcebook of Ideas, edited by David H. Ahl. Creative Computing Press, Morristown, NJ. 222 pages, paperback $15.95. 1979.

According to the back cover, this collection of 77 articles from *Creative Computing* is a sourcebook of ideas for using computers to learn about mathematics.

The ten sections are on Introducing the Computer and How to Buy a Microcomputer System; Thinking Strategies and How To Solve Problems; Computer Simulations; Probability; Mathematical Miscellany (progressions, sports predictions, double precision, circular functions, etc.); Art, Graphics and Mathematics; Computer Assisted Instruction; Programming Style; Short Programs; Puzzles, Problems and Programming Ideas.

The book has over 200 problems for assignment, and nearly 100 programs, nearly all in Basic.

My favorite articles are on computing factorials, solving alphametric puzzles (SEND+MORE = MONEY), multiprecision multiplication, double precision, circular functions, trig patterns on printers, "How to Hide Your Basic Program," and several of the short programs.

This book can be recommended, not only for teachers of computer science and mathematics, but for anyone interested in computer math. The reprinted articles are the best to have appeared in this magazine, from the beginnings in 1974 up to 1979.

From Dits To Bits: A Personal History of the Electronic Computer, by Herman Lukoff. Robotics Press, Box 555, Forest Grove, OR 97116. 230 pages, hardcover $12.95. 1979.

This is the autobiography of a computer pioneer who, until he died last year, was director of technical operations at Sperry-Univac. It is mostly about his involvement in five of the first computers: ENIAC, EDVAC, BINAC, UNIVAC I, and LARC.

After two chapters on his early interest in amateur radio (from which comes the "dits" of the title), Lukoff describes his days as an EE student at the University of Pennsylvania's Moore School. During his senior year, in 1943, he was hired to work on "Project PX" at The Moore School, working for J. Presper Eckert on what eventually became ENIAC.

After two years in the Navy, Lukoff returned to The Moore School to work on a follow-on contract to ENIAC, a computer project called EDVAC, for which, among other things, he devised the control system for the mercury-tank memory.

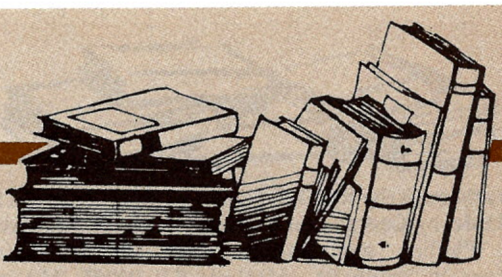
Eckert and John W. Mauchly (who both wrote introductions for this book) then formed the Electronic Control Co., which became the Eckert-Mauchly Computer Corp., which became a division of Ramington Rand, which became Remington Rand Univac, a division of Sperry Rand Corp.

Lukoff stayed on through the various acquisitions and mergers, working on the now-famous computers. He was chief engineer on the LARC program.

In his later years he returned to amateur radio, and bought a Radio Shack TRS-80 computer, which in a photograph caption he describes as being "more powerful than the UNIVAC I central processor."

If you have any interest in the pioneering days of computers, this is a fine book to read, full of personal detail, enough technical details to be interesting but not so many as to be boring, and written with great enthusiasm.

For the non-technical, the book ends with an eight-page glossary.



32 Basic Programs for TRS-80 (Level II) Computer, by Tom Rugg and Phil Feldman. Dilithium Press, Box 92, Forest Grove, OR 97116. 282 pages, paperback $15.95. 1980.

Whether or not you like the mix of programs, this book is a nodel of how such a book should be written and published, and as such is one of Dilithium's best.

For each of the 32 programs, the authors provide sections on Purpose, How to Use it, Sample Run (photographs of the screen, usually), Program Listing, Easy Changes, Main Routines (what the various parts of the program do), Main Variables, and Suggested Projects. The listings and runs are all printed quite clearly.

As for the programs themselves, they are in six groups: applications (biorhythm, checkbook balancing, loan payments, etc.), educational (math drills, metric conversion, vocabulary expansion, etc.), games (a Mastermind lookalike, obstacle race, Wari, etc.), graphics (kaleidoscope and three others), mathematics (least-square curve-fitting, integration, simultaneous equations, etc.), and miscellaneous (approximation of pi, powers of integers, etc.).

The mix is about as good as can be expected, intended to appeal to the widest number of prospective readers, and is much better than several other mixes available in similar books.

What may be unique to this book are the "Easy Changes," which show how to make the program work differently. "You do not have to understand how to program to make these changes," the introduction says. The biorhythm program, for instance, can display the number of days between any two dates if a new line is inserted, and the number of days of the chart shown on the screen can be changed by changing the number in line 360.

Structured Basic And Beyond, by Wayne Amsbury. Computer Science Press, 9125 Fall River Lane, Potomac, MD 20854. 325 pages, paperback $10.95. 1980.

This book is used in an Introduction to Computer Science at Northwest Missouri State University, where the author is an associate professor.

The three-semester-hour course covers Chapters 1-6, on The Basic Machine, Loops and Structures, Names and Messages, Three Data Structures, Expressions and Functions, and Files. Chapters 7-9, according to the preface, are "included for use in a course with a faster pace, for enrichment, and for self-study." They are on Strings and Linked Lists, Stacks and Queues, and Tree-Like Structures.

Appendix A contains answers to the 9 or 10 self-review exercises in each chapter. Appendix B contains a Basic standard, which is actually some comments on an interim report. A good bibliography provides references to books on languages, data structures, simulation, hardware, etc.

The opening words set the style for the book, "A computer spins its wheels — very, very rapidly. Its wheels are really electronic circuits which go through a cycle of operations again and again . . . "

The writing is conversational, the coverage thorough, and the longer programs are fairly easy to understand. The book is intended for interactive terminal use.

Amsbury uses pseudocode to describe the intermediate steps in deriving an algorithm, up to page 52 in conjunction with flowcharts; without them afterwards. This apparently unique feature may dismay flowchart-oriented readers, with the use of "endwhile," "dequeue" and "endif." But it's not too hard to figure out and helps learn how to develop "a clean logical program structure," as Chapter 1 says.

Pathways through the ROM: Guide to Level II Basic and DO Source Code, by George Blank, Roger Fuller, John Hartford John T. Phillipp, and Robert M. Richardson. SoftSid Publications, 6 South St., Milford, NH 03055. 119 pages paperback $19.95 plus $1 shipping. 1980.

As the preface notes, this book is a compilation of fou items previously published separately. Part I consists of th forward, introduction and the first nine chapters o Richardson's $10 *TRS-80 Disassembled Handbook,* an teaches the use of the subroutines residing in the Level II Basi ROM.

Part II is Fuller's $18.95 *Supermap,* which lists, i sequential order, comments on the contents of the Level II Basi ROM, indexed to hex memory locations.

Part III provides two programs from *PROG/80* magazine Philipp's HEX-MEM monitor (written in Basic, permit examining the ROM directly and experimenting with machine language programming); and Blank's Z-80 disassembler, whic when used with a computer, a printer and the comments i Supermap, permit producing a commented source listing of th Level II Basic ROM for personal use.

Part IV contains Hartford's comments on disk operatin systems TRSDOS and NEWDOS, and Western Digital's spe sheet for the floppy-disk controller chip used in the TRS-80

The preface says "These programs, while very useful, wi not satisfy the serious programmer, who will desire mor powerful programming tools," such as Radio Shack's Edito Assembler, a machine-language monitor such as T-BUG, et However, they do provide a great deal of information, pulle together from six sources, at less than 60 percent of the origin cost, that is indispensable to any and all adventuresom explorers of the TRS-80's Level II Basic ROM.

Inside Level II: A Programmer's Guide to the TRS-80 ROMs by John Blattner and Bryan Mumford. Mumford Micr Systems, Box 435-B, Summerland, CA 93067. 70 pages paperback $15.95 plus .75 postage. 1980.

Written mainly for the assembly-language programmer this slim and highly useful volume shows how to use the man sophisticated and useful routines already resident in the ROMs

Part I describes the locations of these routines, th optimum entry points for minimizing their calling sequences and the setups and assembly-language instructions required t call them.

The 10 chapters of Part I are on Key Locations and Entr Points; Registers, Buffers and Variable Passage; Conversio Routines; Arithmetic Operations; Mathematical Functions Keyboard Input; Cassette Input/Output; Video Output; Vide Display Control; and Miscellaneous Routines and Information.

Part II, Linking Assembly Language and Basic Programs shows how to write a single, smoothly-joined program tha combines the best features of both languages, according to th preface. The authors present a program format tha incorporates the convenience and string-manipulating abilitie of Basic with the speed and efficiency of machine-cod subroutines. Instructions are given for creating a composit program structure that loads with the SYSTEM command, an executes in both Basic and machine language.

The preface notes that "To take full advantage of th information in this book requires a knowledge of Z-80 assembl language programming. Readers with such knowledge who ar also fortunate to own a TRS-80 without a disk shoul experience little difficulty in putting this information t immediate use. The presence of a disk complicates matters principally because the Level II ROM has exits to Disk Basi that are not always plugged up when the latter is not present

and also because there are (and will be) different versions of disk-system software. Nevertheless, all the material of this book can be used with disk systems provided sufficient care is taken, and the book details exactly where and how special allowances for a disk need to be made."

The 8086 Primer: An Introduction to Its Architecture, System Design and Programming, by Stephen P. Morse. Hayden Book Co., Rochelle Park, NJ. 214 pages, $8.95 paperback. 1980.

Although the text doesn't tell you, the back cover notes that the author is "the man responsible for the architectural definition of the 8086 processor."

After an introductory chapter you can skip if you have a good knowledge of computers and small computers, the book describes 8086 architecture, going into machine organization (register and memory structure, addressing modes) and the 8086 instruction set.

Chapter 4, on system design, considers the 8086 as a circuit component, and shows how to use it, together with other components, to form a complete small computer system.

Programming is divided into assembly-language programming using the low-level ASM-86, and 8086 high-level-language programming using PL/M-86, in the last two chapters.

Morse writes very clearly, covers a lot of territory, and uses very helpful illustrations that in some cases are uniquely ingenious.

Although the book is of course most easily understood by those already familiar with at least one microprocessor, the author's easygoing style permits even a bright beginner to get a great deal of information from this very well written book.

All About Personal Computers. Datapro Research Corp., 1805 Underwood Blvd., Delran, N.J. 08075. 65 pages, paperback $25. 1980.

Here is a wealth of detailed information about 15 personal computers, sandwiched in between five pages of so-so introductory material and 22 pages of listings of vendors, of little use since the listings contain only addresses and phone numbers.

However, Section 2 gives more details on the 15 systems than you're liable to find anywhere else, including the manufacturers' own spec sheets. Each computer gets two pages: the first provides a photograph, background information, and system characteristics. The second page gives full details on hardware and packaging, software, and support services (documentation, support, terms and conditions).

Section 1, All About Personal Computers , discusses the history of personal computers (briefly), current and projected market sizes, current applications, how personal computers are sold (stores, dealers, computer companies), future trends, and How to Buy a Personal Computer (evaluate your needs, identify which systems seem to satisfy your requirements, try them hands-on, make your selection).

Section 3, Directories , seems mainly a filler, providing no information at all about product, only the addresses and phone numbers of hardware, software and peripheral vendors. Many of the companies listed under Personal Computer Peripherals Vendors are highly unlikely, such as CPT, Data General, DEC, Wang, Control Data, etc., page after page of interesting but almost entirely useless listings.

But for the serious beginner, or even for the expert, who wants all the data he can get on the top 15 personal computers, this is the thing to buy.

(Datapro publishes detailed, looseleaf, updated reports on software, minicomputers, office automation, etc., and a variety of short reports.) □

Instructions

To assemble *The (Not So) Concise Pocket Dictionary of Computer Terminology* simply follow these steps:

1. Cut on dotted lines.
2. Assemble pages in order with cover on top.
3. Bind at top edge or left edge (but not both). To bind, use staples, yarn or heavy thread.
4. Carry with you so you can instantly decipher computer terms when you see them in *The Chronicle* on matchbook covers, or on subway ads.

The (Not So) Concise Pocket Dictionary of COMPUTER TERMINOLOGY

by
Peter Payack
&
Opal Louis Nations

A

ALGORITHM: Trendy dance for hip programmers.

ANALOG: A roughly hewn wooden seat, used by computer analysts.

APPLICATION: The act of applying or laying on of color decals to hardware surfaces.

B

BASIC: Acronym for **B**unglers **A**ll-purpose **S**lang **I**nterjectory **C**laptrap, a series of systematized exclamations taught to initiates.

BUG: (See coding).

1

C

CARD READER: Tarot technician.

COBOL: An acronym for a computer language that is **C**ompletely **O**ver and **B**eyond reason **O**r **L**ogic.

CODING: A cough syrup for treating a sick computer who caught a bug. (See bug).

D

DATA PROCESSOR: A mechanism which chops, shreds, slices, or grates information into digestible bits.

DATA, TERMINAL: Autopsy report.

DEBUG: Delicing the computer center.

2

DIGITAL COMPUTER: A machine for processing fingerprints.

DISC PACK: A six pack of alcoholic beverage, designed for consumption aboard an alien space craft.

F

FORTRAN: Short for Formula Training, a nourishment given to baby (or micro) computers.

G

GENERAL Purpose Computer: A four star all purpose war gaming machine.

H

HEAVY PENCIL: (See Light Pen).

3

I

INPUT: CENSORED (See Input Device).

INPUT DEVICE: A male electronic probe designed for computation, and procreation of future micro computers.

K

KEYBOARD: A dozing computer programmer.

L

LIGHT PEN: See Heavy Pencil.

LINE PRINTER: A stand-up gag writing computer for composition of one-liners.

N

NANO-SECOND: A second in Mork Time.

4

O

OUTPUT DATA: Data that has been excreted from an internal storage medium to a flush-down device.

P

PAPER TAPE: A long, paper-thin worm that lives as a parasite in the intestines of a storage medium.

PRINTER: Johann Gutenberg (1398? - 1468).

PROGRAM: (Short for ProGrammy). A music award for professional keyboard operators.

PUNCHED CARD: An inebriated card having almost 80 random columns representing nothing intelligible.

5

S

SOFTWARE: Formal evening attire for female computer analysts.

STORAGE DEVICE: Cold room, or morgue, for Terminal Data.

SYSTEM: A foolproof method for dealing and winning at punched cards.

T

TERMINAL: a) A Grand Central Computer Station for divergent trains of thought.
b) See Data.

U

USER: An addicted individual who requires the services of a methadone processing center.

6

AIR TRAFFIC

CONTROLLER

In **Air Traffic Controller** you assume responsibility for the safe flow of air traffic within a 15x25 mile area up to 5,000 feet in altitude. During your shift as a controller in charge of this airspace, 26 aircraft become active and under your control. Jets and prop planes have to be guided to and from the two airports, navigational beacons and ten entry/exit fixes. The aircraft enter the controller's airspace at various altitudes and headings whether or not you are ready.

Air Traffic Controller retains the basic realism of air traffic control. This program requires the same steady nerves under pressure and the same instant, almost instinctive, analyses of complex emergencies which are demanded of a professional air traffic controller. But "ATC" adds the excitement and well-defined goals of a game. This is just a simulation, and all passengers left in air traffic limbo by a panicked player will live to fly another day.

Your goal is to get all of the aircraft to their assigned destination before the shift is completed. At your disposal are a radar display of the aircraft positions in the control area; coded information concerning aircraft heading, destination and fuel supply; navaids enabling you to hold aircraft or assign them automatic approaches; and commands to alter the altitude and heading of the aircraft. Working against you are altitude and heading requirements, fuel restrictions and, of course, the inimitable clock.

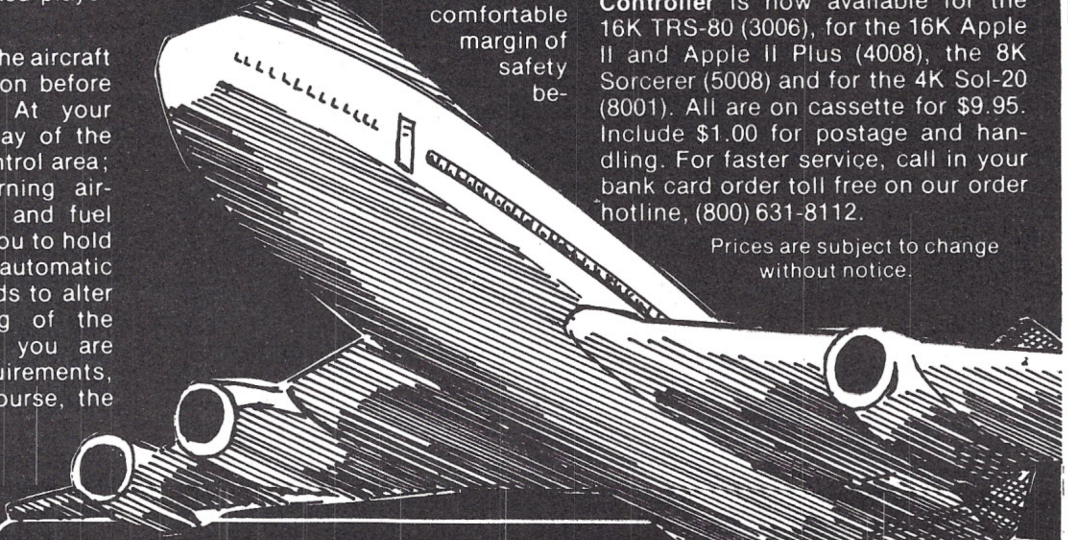
The most obvious measure of difficulty of a game is the clock setting at the beginning. In a 99 minute game you will have time to go fix a sandwich between the appearance of two successive aircraft, while in the 16 minute game you may not have time to swallow before all of the aircraft have appeared.

No two games, even at the same clock setting, are ever alike. As controller, you must cope with the unique requirements of each aircraft. The game will end if you commit a "boundry error," that is, if an aircraft fails to leave your area at the proper altitude and exit fix...causing an unpleasant surprise for the controller next door. The game also ends if you fail to leave a comfortable margin of safety between the aircraft as they whiz past each other. In cases of excessive delay, fuel supply considerations will become invested with a particular sense of urgency.

Successful guidance of all aircraft to their destination is a heady accomplishment. This never fails to thrill ATC enthusiasts at each successive level of play.

Your local retail store should carry Creative Computing Software. If your favorite retailer does not carry the software you need, have him call in your order to (800) 631-8112. Or you can order directly from Creative Computing. Write to Creative Computing Software, P.O. Box 789-M, Morristown, NJ 07960. **Air Traffic Controller** is now available for the 16K TRS-80 (3006), for the 16K Apple II and Apple II Plus (4008), the 8K Sorcerer (5008) and for the 4K Sol-20 (8001). All are on cassette for $9.95. Include $1.00 for postage and handling. For faster service, call in your bank card order toll free on our order hotline, (800) 631-8112.

Prices are subject to change without notice.

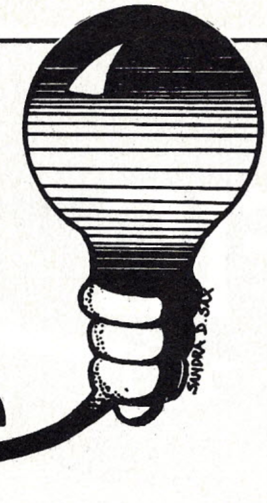


creative computing software

sensational software

CIRCLE 300 ON READER SERVICE CARD

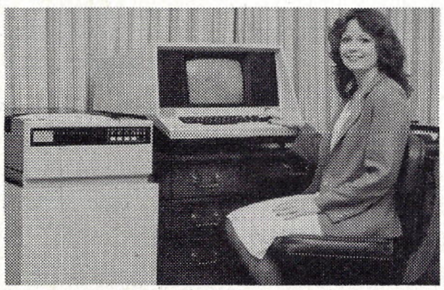
Compleat Computer Catalogue



COMPUTERS

64K TABLE TOP SYSTEM

SD Systems announces the SD-700, a 64K table top computer system that includes a high speed 32mb fixed/removable disk. The disk can be expanded to a maximum of 96mb.



The SD-700 is a single or multi-user system. In its multi-user configurations, it supports up to five users with user partitions of 48K and a 16K operating system with a maximum memory capacity of 256K.

The standard configuration offered provides two Z-80 central microprocessors operating at a 4 MHz CPU speed and two Input/Output ports. Included with the SD-700 is Cosmos, a Communications Oriented Multi-user Operating System which supports CP/M compatibility, Cobol, MBasic, CBasic, and other compilers, and individual and shared files for multi-user applications.

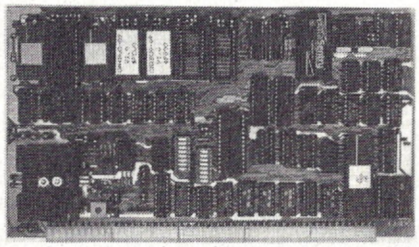
SD Systems, P.O. Box 28810, Dallas, TX 75228. (214) 271-4667.

CIRCLE 301 ON READER SERVICE CARD

6809 SYSTEMS

Gimix 6809 Systems feature 25 Amp. c.v. ferro-resonant transformer, fifteen 50- and eight 30-pin bus slots, a minimum of 32K of Static RAM, and a choice of I/O cards. A variety of system monitor options are available including the GMXBUG 09 monitor/debugger and SWTP's SBUG-E monitor.

Gimix' 6809 CPU is an SS-50 proces-



sor board which features selectable processor clock speeds of 1, 1.5, and 2 MHz. It has provisions for a variety of on board devices, including a 9511 or 9512 arithmetic processor, 6840 programmable timer, time of day clock with battery back-up, 1K of scratchpad RAM which can be CMOS with battery back-up, and four PROM/ROM/RAM sockets that can hold up to 32K of on board software.

Gimix, Inc., 1337 West 37th Pl., Chicago, IL 60609. (312) 927-5510.

CIRCLE 302 ON READER SERVICE CARD

SMALL BUSINESS COMPUTER SYSTEM

The MCS 7000 is a self-contained desk-top microcomputer system with a built-in keyboard, high-resolution CRT display, 64K capacity, and two dual-sided floppy discs. Available with the system is a high-speed printer that doubles as an office typewriter.

It features an RS-232C port and can be interfaced with standard TTY's, "dumb" terminals, other MCS 7000 computers, or tied into a main frame computer. Standard programs include: Accounts payable, accounts receivable, mailing lists, payroll, word processing, inventory control, general ledger, and sales charts.

MCS 7000 has a full ASCII keyboard, 9-key cursor, a 12-key numeric entry keypad, and 15 special keys for custom symbols. The CRT display is a 12" screen, 24 characters deep by 80 characters across, and features reverse video, half tone, field, character blinking, and addressable cursor. Prices range from $4995 to $9995.

Micro Card Systems, Inc., 185 Main St., Port Washington, NY 11050.

CIRCLE 303 ON READER SERVICE CARD

OFFICE SYSTEMS

CMC Marketing introduces a new line of computer systems. These include the Series 100 designed for the small business and word processing user and the Series 200, a multi-terminal system designed for use in larger businesses with expanded data processing requirements.

The systems are built around the Z80A microprocessor and feature the industry standard S-100 bus mainframe. Systems can be configured with diskette storage of up to 4 megabytes and hard disk storage of up to 28 megabytes.

All CMC systems utilize the CP/M operating system, enabling user to run most CP/M software on the system.

CMC Marketing Corp., 10611 Harwin, Suite 406, Houston, TX 77036. (713) 995-4960.

CIRCLE 304 ON READER SERVICE CARD

MULTI-USER MICROCOMPUTER SYSTEM

Intertec Data Systems Corporation has announced a multi-user "shared-disk" microcomputer system called the CompuStar.

The CompuStar system consists of a network of video display terminals which employ their own internal microprocessor

and dynamic RAM. The terminals are tied together in a "network" fashion to "share" the resources of a single Winchester or other hard disk device.

The system architecture is based around one of three Disk Storage Systems consisting of a hard disk device, complete with power supply, and a disk controller and multiplexor circuitry to "tie" user stations into a common disk system. A tabletop 10 megabyte Winchester-type drive is offered which will retail for $3995.

Intertec Data Systems, 2300 Broad River Rd., Columbia, SC 29210, (803) 798-9100.

CIRCLE 305 ON READER SERVICE CARD

LOW COST SINGLE- BOARD COMPUTER

The Model SBC-02 single-board computer is a minimal 4-chip system on a 6″ x 6″ printed circuit board, which features a 6802 processor with 128 bytes of RAM, 2K of EPROM, and parallel or serial I/O. A wire-wrap area is provided for custom interfacing or other expansion.

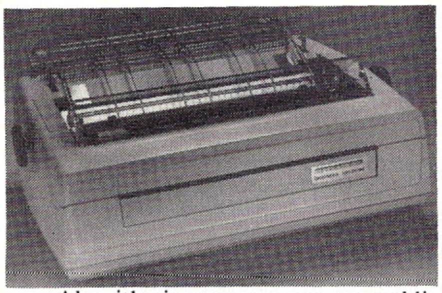
A machine level monitor called Humbug can be installed to provide program entry and control, single-stepping, breakpoints, and other front-panel functions from a serial terminal. In single quantities, the computer costs $25 for a bare board with instructions, $75 for a parallel I/O kit, or $150 when wired and tested.

Star-Kits, P.O. Box 209, Mt. Kisco, NY 10549.

CIRCLE 306 ON READER SERVICE CARD

TERMINALS & I/O

SCIENTIFIC WORD PROCESSING SYSTEM



Algorithmics announces a new addition to its line of ALGO-2100 word processing systems that is designed to produce scientific manuscripts, journals, and reports which contain equations.

The system is based on the standard ALGO-2100 but includes a special extended character printer, system modifications to display special characters on the screen, and keyboard modifications that allow the operator to type scientific and Greek characters along with regular text.

The scientific printer is a letter-quality impact device with over 45 different type styles on interchangeable molded thimbles similar in appearance to a badminton birdie. Printer speed is 55 characters per second even when printing scientific notation.

Multi-level equations that might involve several layers of numerators, denominators, and superscripts can be typed in a precision, expanded vertical layout. Each equation line is typed separately, and the operator instructs the system how far to space down for the next line in increments as fine as 1/48 of an inch. Unlimited numbers of superscripts and subscripts are allowed.

The new scientific system is completely compatible with other Algo-2100 systems, and is available as an upgrade to existing customers.

Algorithmics, Inc., 177 Worcester Rd., Wellesley, MA 02181. (617) 237-7226.

CIRCLE 307 ON READER SERVICE CARD

VOICE I/O TERMINAL FOR SORCERER

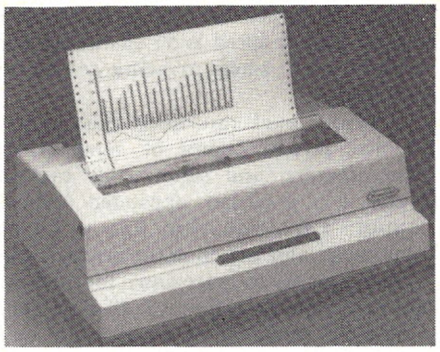
Cognivox plugs into Exidy's Sorcerer computer and offers a 16-word recognition vocabulary plus voice response with up to 16 words or phrases. Recognition accuracies of up to 98% are possible with cooperative speakers.

Software available includes Voicetrap, a voice operated video game and Vothello, a voice input version of the popular game Othello. A talking calculator program allows the Sorcerer to be used as a four function calculator without looking at the CRT screen, and a vocal memory dump program can read its memory out loud in Hexadecimal format. $149.

Voicetek, P.O. Box 388, Goleta, CA 93017.

CIRCLE 308 ON READER SERVICE CARD

COLOR GRAPHICS PRINTER



The PrintaColor IS8001 is a color printer terminal, with an auxiliary computer within the terminal itself. It consists of a three-color ink jet print head that is capable of giving graphic output, and internal electronic circuitry which includes the computer. The computer within the terminal enables the unit to interface efficiently with the host computer, so that the unit may operate with minimal burden on the host computer's processing capacity.

The printing head has 12 ink nozzles and is capable of printing any of seven colors. Each of the ink colors has 4 nozzles, and the color ink jet head has a resolution of 100 dots per inch horizontally and vertically. $6000.

PrintaColor Corporation, Norcross, GA. **CIRCLE 309 ON READER SERVICE CARD**

TRS-80 CASSETTES LOAD AT HIGH SPEED

Personal Micro Computers, Inc. has announced a device to input pre-recorded programs into TRS-80 Level II computers at 16 times normal speed. Standard cassettes can now be loaded at 8000 baud using a modified CTR-41 recorder and the Fastload Cassette Interface.

Any cassette program previously "saved" at normal speed can be loaded at high speed. Fastload is also capable of searching at high speed for Basic programs by a single character designation or for systems programs by a name of up to six characters in length. $188.

Personal Microcomputers, Inc., 475 Ellis St., Mountain View, CA 94043.

CIRCLE 310 ON READER SERVICE CARD

CHESS PRINTER

Fidelity Electronics, Ltd. has introduced the Fidelity Challenger Printer. Intended to be a companion for the Voice Sensory Chess Challenger, the Challenger Printer records and numbers both black and white moves on a single line.

Utilizing standard, commercially available 2¼″ thermal paper, the Challenger Printer displays current board positions by printing graphic display of black and white pieces in their actual locations.

Fidelity Electronics, Ltd., 8800 NW 86th St., Miami, FL 33178.

CIRCLE 311 ON READER SERVICE CARD

MUSIC & GRAPHICS

GRAPHIC TERMINAL FOR NORTHSTAR

Sigma Computers announces the 1042S high resolution, high performance graphics terminal for North Star, which features a 15-inch display and memory mapped I/O. The device employs video raster scan technology, constructing graphic images from 640 x 800 dot matrix covering a 7.5″ x 9″ CRT surface and backed up by a 65K bit map display memory.

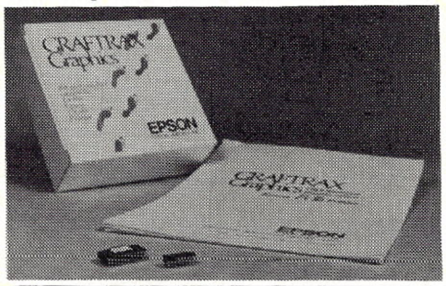
The 1042S has been designed to plug into the S-100 bus and, by virtue of a breakthrough in display memory mapping, provides memory mapped I/O. Retail price is $4,500. including CRT monitor, Controller, and Display memory on 2 printed circuit boards with S-100 plugs and attractive desk-top packaging.

Sigma Information Systems USA, Inc., 556 Trapelo Rd., Belmont, MA 02178. (617) 484-2063.

CIRCLE 312 ON READER SERVICE CARD

GRAPHICS FOR TX-80

Graftrax, a high-resolution, bit plot graphics capability for the Epson TX-80 dot matrix printer, has been introduced by Epson America, Inc.



According to the company, when the bit plot mode is invoked, each bit arriving at the parallel interface individually controls one of seven print wires. The timing is so arranged that each seven-bit word causes the head to print at one dot position for a total of 480 dots per line. No buffer is used in the printer.

Utilizing a PROM, GRAFTRAX also enables the TX-80 to perform programmable U.F.H. (Universal Forms Handling) functions. The length of a line feed is software definable in 255 steps of .007″ each.

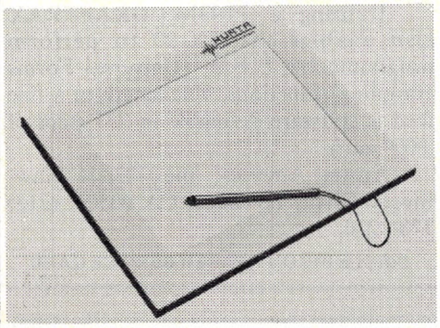
Epson America, Inc., 23844 Hawthorne Blvd., Torrance, CA 90505. (213) 378-2220.

CIRCLE 313 ON READER SERVICE CARD

CIRCLE 182 ON READER SERVICE CARD

GRAPHIC TABLET

The Kurta Corporation announces a graphic tablet designed for small computers.



Kurta Graphic Tablets are compatible with standard 7″ x 9″ display screens. The output of the tablet directly matches the capabilities of the computer, minimizing both interface and software requirements. Resolution is 100/200 points per inch, and the conversion rate is 100 coordinate pairs per second.

Kurta Corporation, 206 S. River Dr., Tempe, AZ 85281. (602) 968-8709.

CIRCLE 314 ON READER SERVICE CARD

IMPROVED GRAPHICS FOR OSI

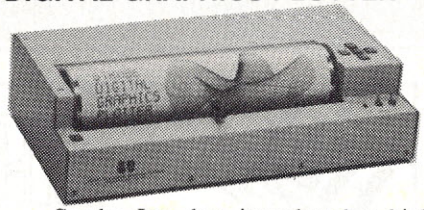
Hobbyware has developed a simple, plug-in, mod to quadruple the vertical resolution and double the horizontal resolution of any OSI computer.

The credit-card sized board is installed in place of a chip which is then installed into the mod board. One wire must then be connected as an enable line. With a simple timing adjustment, the computer is ready to go with a complete graphics capability of 128 x 128.

Hobbyware, 838 Parkside, Elmhurst, IL 60126.

CIRCLE 315 ON READER SERVICE CARD

DIGITAL GRAPHICS PLOTTER



Strobe Inc. has introduced a high performance digital graphics plotter. The Strobe Model 100 is a drum type plotter utilizing four-phase stepping motors to provide drum rotation and linear pen motion.

Its step-size, the smallest motion the plotter can make, is .004 inch along each axis.

Strobe provides assembly language vector software support for 8080/8085, Z80, and 6502 microprocessors.

The Model 100 can be interfaced to any computer through two parallel 8-bit output ports and one 8-bit input port. Optional interfaces for the TRS-80, Apple II, CBM PET, and S-100 bus computers are available. Also offered is a plot applications software package that runs with most versions of Basic and Fortran. $680.

Strobe Inc., 897-5A Independence Ave., Mountain View, CA 94043. (415) 969-5130.

CIRCLE 316 ON READER SERVICE CARD

DIGITIZER FOR APPLE

The DT-11A digitizer for the Apple II version of the Hi Pad digitizer provides a slot interface card for the Apple II, a floppy based software package, menu overlay, and stylus.

Functions supported by the software

include draw, line, area, background, pen color, separate, catalog, save, load, shape, etc. A plastic overlay that serves as a menu allows the user a selection of these functions and gives the user the opportunity to generate a wide variety of creative color graphics.

Besides the interface and software, the DT-11A offers slot independence, Basic and Pascal compatibility, assembler driver code, user controls, and optional cursor. The DT-11A/Apple II system requires a 48K system and the Applesoft firmware card. $795.

Houston Instrument, One Houston Square, Austin, TX 78753. (512) 837-2820.

CIRCLE 317 ON READER SERVICE CARD

PERIPHERALS

COMMUNICATIONS LINK

Digicom Data Products, Inc. has introduced an intelligent communications controller designed for interconnection between various terminals and modem devices.

Called Digilink," it offers text editing through an assigned terminal port, interactive communications through a modem control port and data collection via the printer control port. When available, the optional firmware features will provide async/sync conversion, code conversion, serial to parallel connection, IEEE-488 interface plus various terminal protocol emulation packages.

Depending on user requirements and configuration the Digilink can be used in applications ranging from off-line text editing, limited word processing, station to station electronic mail, terminal data concentration and dual station multiplexor. $995.

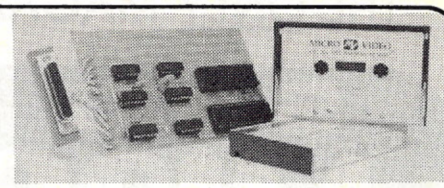
Digicom Data Products, Inc., 1440 Knoll Circle, San Jose, CA 95112. (408) 279-8711.

CIRCLE 318 ON READER SERVICE CARD

RS-232C INTERFACE FOR INTERACT

Micro Video has released an RS-232C peripheral interface for the Interact computer, along with printer and communications software.

The interface is equipped with a dual port that has handshaking and send/receive capabilities for driving any RS-232-compatible device. All I/O parameters are software-selectable from Basic and machine code.

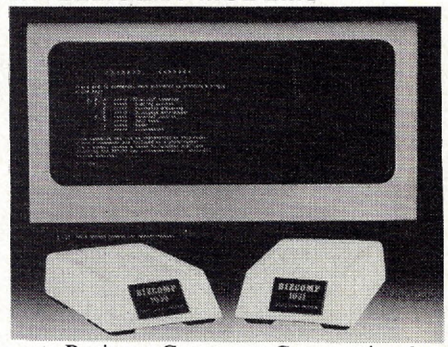


The RS-232 Pack includes Microsoft Basic with printer access commands and a Basic editor with automatic line numbering, resequencing, string substitution, appending, and other useful functions. Communications software is available separately. $129.95.

Micro Video, 204 E. Washington, Ann Arbor, MI 48104.

CIRCLE 319 ON READER SERVICE CARD

INTELLIGENT MODEMS



Business Computer Corporation has introduced Bizcomp Model 1030 and 1031 Intelligent Modems.

The 1030 Series combines a low error rate modem with an automatic calling unit and custom BIZ-080 microcomputer in a compact FCC-registered unit with auto-answer, auto-dial and auto-repeat dial features.

Interfacing to RS232-equipped computers, terminals and word processors requires only a 3-wire data cable. A current loop interface is also standard. $395.

The 1031 adds command-selectable dial pulse or tone dialing, and self-test. $495.

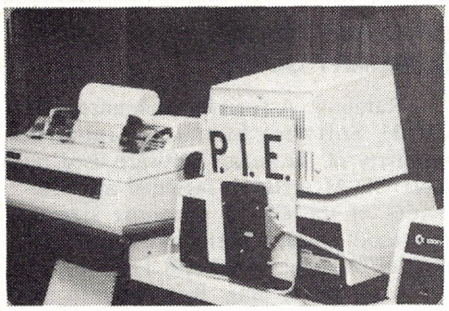
Business Computer Corporation, P.O. Box 7498, Menlo Park, CA 94025. (415) 854-5434.

CIRCLE 320 ON READER SERVICE CARD

IEEE-488 TO PARALLEL INTERFACE FOR PET

LemData Products introduces the P.I.E.-C; a full featured, IEEE-488 to parallel interface for the PET/CBM computers.

Because the P.I.E.-C has parallel output with two handshaking lines it is compatible with the Centronics printers, NEC Spinwriter, Anderson-Jacobson AJ841, Integral Data System's "Paper Tiger," Anadex 8000 and 9000 type printers, as well as other parallel-input ASCII printers.



The P.I.E.-C can respond to any IEEE-488 primary address by setting its DIP switch. The conversion of non-standard PET/CBM codes to true ASCII codes is also switch selectable. $119.95.

LemData Products, P.O. Box 1080, Columbia, MD 21044. (301) 730-3257.

CIRCLE 321 ON READER SERVICE CARD

VENDOR LITERATURE

DIRECTORY OF EDUCATIONAL SOFTWARE

Queue's Catalogue #3 is a directory of educational software available for Apple, Pet, TRS-80 and Atari. Programs from over 40 educational software publishers are grouped by computer, subject matter and grade level, and described.

All the programs can be ordered directly through Queue. $8.95.

Queue, 5 Chapel Hill Dr., Fairfield, CT 06432.

CIRCLE 322 ON READER SERVICE CARD

BOOKS AND BOOKLETS

TRS-80 SOFTWARE CATALOG

Racet Computes announces a 24-page catalog of software for the TRS-80. Included are descriptions of all new products including: Hard/Soft Disk System, hard disk drive interface software for the Mod II; Spoolers; Basic Cross Reference Utility for the Mod II; and Blink, a utility for linking Basic programs to each other.

Racet Computes, 702 Palmdale, Orange, CA 92665. (714) 637-5016.

CIRCLE 323 ON READER SERVICE CARD

BUSINESS MICROCOMPUTER EVALUATIONS

MIC, a data processing publication company, has prepared objective evaluations of the following 12 business microcomputers: Radio Shack TRS-80, Model I and II; IBM 5110 & 5120; Vector Graphic Systems;

Alpha Micro; Ohio Scientific; Rexon; North Star; Cromemco; Apple II and Burroughs B90.

Each of the evaluations is an in-depth analysis of the computer product, its hardware, and software. The set of business evaluations is available for $75.

MIC, 140 Barclay Center, Cherry Hill, NJ 08034. (609) 428-1020.

CIRCLE 324 ON READER SERVICE CARD

WORD PROCESSING REPORT

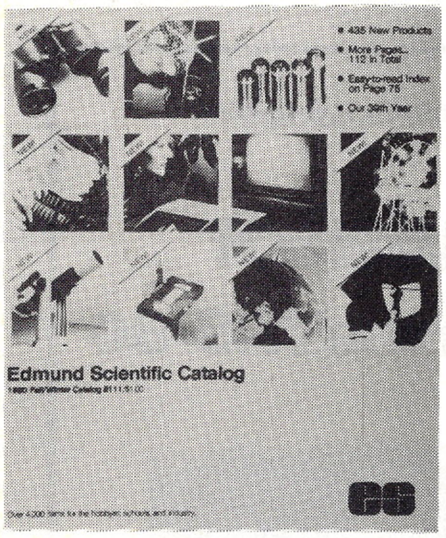
The Small Systems Group has begun publication of a series of in-depth product evaluation reports. The first of these reports, entitled "Word Processing on Personal Computers" is now available.

The report introduces personal computer word processing with sections on software, hardware and applications. It continues with general descriptions of four programs: Auto Scribe, Electric Pencil, Magic Wand and Wordstar. These are compared on quality of documentation, ease of learning, editing power and formatting power. $10.

Small Systems Group, Box 5429, Santa Monica, CA 90405.

CIRCLE 325 ON READER SERVICE CARD

EDMUND SCIENTIFIC CATALOG



The Edmund Scientific Fall/Winter catalog for 1980 introduces 435 science-related products for hobbyists and experimenters.

A new section in the catalog, called "Products for Scientific Living" features a limited edition space lithograph, a computerized thermostat, a unique speech controller cassette player and other unusual items designed for contemporary life styles, and several new energy-saving products have been added to the line of solar products.

Edmund Scientific Co., 7082 Edscorp Bldg., Barrington, NJ 08007.

CIRCLE 326 ON READER SERVICE CARD

CP/M SOFTWARE SUMMARY GUIDE

Rainbow Associates announces the CP/M Software Summary Guide, a summary of the major software used on most CP/M systems. Included are summaries of the CP/M operating system, Microsoft Basic, CBasic, and the CP/M utilities Despool, MAC and TEX.

Sixty pages long, it is designed for easy use. Features are organized alphabetically, so the reader can find an explanation quickly. $3.75.

Microsystems Book Service, P.O. Box 789-M, Morristown, N.J. 07960 (201) 267-4558.

CIRCLE 327 ON READER SERVICE CARD

NEWSLETTERS

NEWSLETTER FOR TEACHERS

Teaching Computer Programming is a monthly newsletter for instructors at junior high school and high school levels. Though there will be articles for TRS-80 and Apple II, the content will be kept machine-independent whenever possible.

Topics to be covered include programming assignments, quiz ideas, programming techniques, and uses of computers and peripherals in the classroom. A one-year subscription is $8.

Craig Nansen, 1112 Glacial Dr., Minot, ND 58701.

CIRCLE 328 ON READER SERVICE CARD

EDUCATIONAL NEWSLETTER

A monthly newsletter, *Microcomputers in Education,* commences publication in October. The newsletter will carry reviews of educational software, new product announcements, reports on CAI in the classroom, reviews of books and magazine articles, news of meetings, and industry news.

Yearly subscriptions are $15.

Microcomputers in Education, 5 Chapel Hill Dr., Fairfield, CT 06432.

CIRCLE 329 ON READER SERVICE CARD

SINCLAIR NEWSLETTER

Syntax ZX80, a monthly newsletter for Sinclair ZX80 friends and owners is devoted to news and reviews of ZX80 hardware and software.

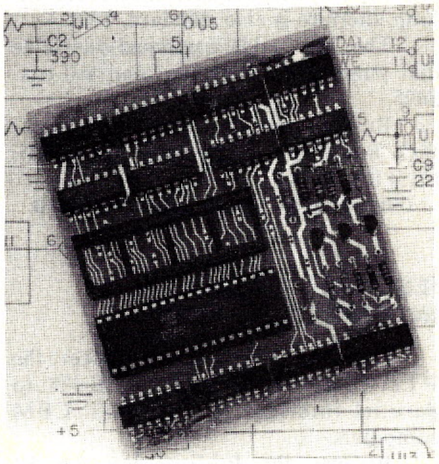
Syntax ZX80 will also provide readers with forecasts of hardware and software, applications, and technical details for do-it-yourselfers, as well as a forum for users to share advice about programs and vendors. The yearly subscription rate (12 issues) is $25.

Ann Zevnik, Editor, The Harvard Group, Bolton Rd., R.D. 2, Box 457, Harvard, MA 01451.

CIRCLE 330 ON READER SERVICE CARD

DISK & TAPE SYSTEMS

DOUBLE-DENSITY ADAPTER FOR TRS-80



Percom announces a double-density disk controller adapter for the TRS-80. Called the Doubler, it can store up to 345 formatted Kbytes.

A proprietary design allows the user to continue to run TRSDOS, NEWDOS and Percom OS-80 single-density programs without modifying either software or hardware.

The adapter plugs into the controller chip socket of the computer expansion interface. $219.95.

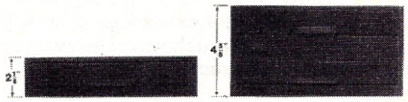
Percom Data Company, 211 N. Kirby, Garland, TX 75042. (800) 527-1592.

CIRCLE 331 ON READER SERVICE CARD

SKINNY FLOPPY

An 8″ floppy disk drive said to be less than one-half the height of any other model has been introduced by Micro Peripherals, Inc.

Height is only 2⅛ inches at the highest point. Other dimensions are length 12½ inches and width 8½ inches.



The new drive, only 2⅛″ high, is available with dual heads (model 82) for reading/writing on both sides of the disk or one head (model 81) for single side operation. Storage capacity is 800 Kbytes for the single-head model and 1,600 Kbytes for the dual.

Micro Peripherals, Inc., 9754 Deering Ave., Chatsworth, CA 91311.

CIRCLE 332 ON READER SERVICE CARD

DISK DRIVES FROM COMMODORE

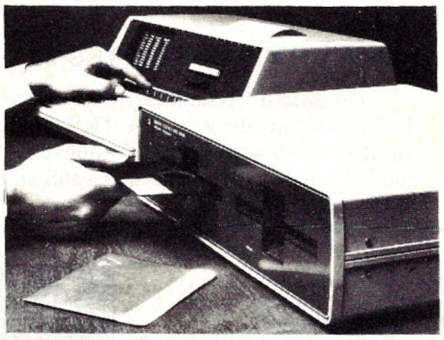
Commodore has introduced the CBM 2030 series of small, light weight 5¼″ floppy disk drives for PET and CBM computers. The single drive model, CBM 2031, is priced at $595 and provides 130K bytes of storage. It may be field upgraded to a dual drive version which offers 260K bytes of storage.

Also available is the CBM 8060 series of high end 8″ floppy disk drives. The CBM 8061, priced under $2500, uses two single-sided drives for 1.6 megabytes of storage; the CBM 8062 uses two double-sided disks for 3.2 megabytes and sells for under $3500.

Commodore Business Machines, Inc., 3330 Scott Blvd., Santa Clara, CA 95051. (408) 727-1130.

CIRCLE 333 ON READER SERVICE CARD

DISK DRIVE FOR HP-85



A flexible disk drive family that provides up to 1.08 Mbytes of on-line capacity for the HP-85 personal professional computer has been introduced by Hewlett-Packard.

The HP 82900 Series drives read double-sided, double-density, 5¼″ disks, and can be configured to provide from 270 Kbytes to 1.08 Mbytes of storage. The interface between the HP-85 and the disk drives is the HP-85 Mass Storage ROM. Prices range from $1300 to $2500.

Hewlett-Packard Company, 1507 Page Mill Rd., Palo Alto, CA 94304.

CIRCLE 334 ON READER SERVICE CARD

8″ FLOPPY DISK SYSTEM FOR H-8 AND H-89

Heath Company introduces an 8″ dual-sided, dual-density floppy disk system. The H-47, designed for use with the H-8 and H-89 microcomputers, provides up to 2½ million bytes of on-line data storage, more than 12 times the current maximum capability.

Featuring two 8″ disk drives as standard equipment, the H-47 Floppy Disk System is fully compatible with current Heath 5¼″ disk systems. Both Heath's HDOS Operating System and CP/M will be supported. Each will permit transfer of data between 5¼″ and 8″ disks.

Heath Company, Dept 350-440, Benton Harbor, MI 49022.

CIRCLE 335 ON READER SERVICE CARD

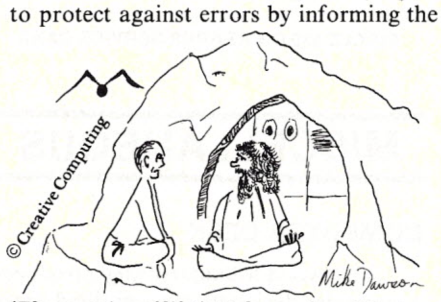
DISK DRIVE FOR TRS-80 MODEL II

Parasitic Engineering announces that its Maxi-Disk 8″ floppy disk drives are now compatible with the TRS-80 Model II. When used with the Model II, Maxi-Disk drives are functionally identical to Radio Shack expansion drives, but provide additional features at a competitive price. No software or hardware changes are needed.

Maxi-Disk drives have activity lights to protect against errors by informing the user when it is safe to remove diskettes. $845.

Parasitic Engineering Inc., 1101 Ninth Ave., Oakland, CA 94606. (415) 839-2636.

CIRCLE 336 ON READER SERVICE CARD



'The meaning of life?' We have released that as a basic program for all micro systems. It's $11.95 at stores everywhere. Have a pleasant return journey.

8″ DISK DRIVE FOR TRS-80, APPLE, S-100

Matchless Systems has announced the MS-800, an 8″ inch disk drive compatible with TRS-80 Models I and II, Apple II and S-100 systems.

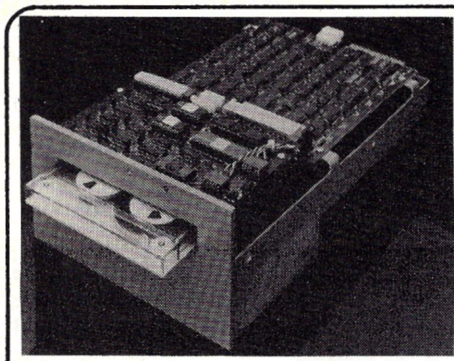
The MS-800 has a capacity of 77 tracks, 26 sectors per track and 128 bytes per sector for a total of 256,256 bytes. The data transfer rate is 256,000 bits per second, and the drives are powered independently of the systems with which they are used. The price range of $995 - $1595 includes all hardware, software and documentation.

Matchless Systems, 18444 South Broadway, Gardena, CA 90248. (213) 327-1010.

CIRCLE 337 ON READER SERVICE CARD

STREAMING 1/4″ CARTRIDGE TAPE DRIVES

Archive Corporation has announced a family of streaming ¼″ cartridge tape drives featuring a recording density of 8,000 bits per inch, and operating speeds of 30 and 90 inches per second.



Designated Sidewinder, the new drives are specifically designed for Winchester disk drive back-up. Utilizing the ANSI-standard 450-foot, ¼″ wide tape cartridge, the drives are offered in either a Basic or Intelligent configuration with 10 megabyte or 20 megabyte capacities.

Sidewinder's erase-write-read recording head operates in a 2 or 4 track format, the tracks being recorded serially using a serpentine recording technique. Prices range from $469 to $954.

Archive Corporation, 3540 Cadillac Ave., Costa Mesa, CA 92626.

CIRCLE 338 ON READER SERVICE CARD

WINCHESTER DISK BACKUP

Tape Interchange Package is an S-100 compatible cartridge tape subsystem and software utility that permits transfer of programs and data files from a Winchester disk to an easy-to-handle 13.4 MB 1/4″ tape cartridge.

Facilitating off-premises data base storage and shipping, and links to the Winchester disk under Digital Research CP/M and MP/M Operating Systems.

With a 2 min./MB backup/restore rate, the Alloy Engineering TIP provides file-oriented backup to optimize cartridge tape utilization. It consists of a DS-100 Controller, 6400 BPI cartridge tape drive, and rack or table mount power supply. Software is distributed on a single sided, single density 8″ floppy disk in CP/M format.

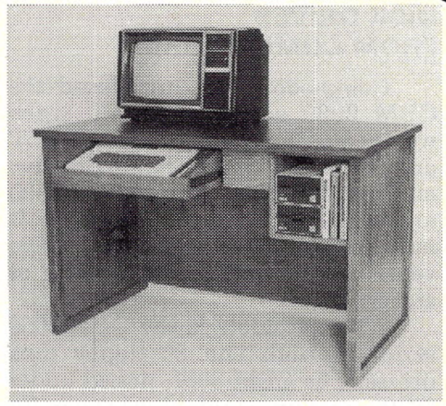
Alloy Engineering Company, Inc., Computer Products Division, 85 Speen St., Framingham, MA 01701. (617) 620-1710.

CIRCLE 339 ON READER SERVICE CARD

MISCELLANEOUS

COMPUTER DESK

Furnwood Manufacturing introduces a computer desk with custom fitted areas to hold the Apple II or Apple II Plus. The drawer in front is designed to hold the Apple Computer at a comfortable typing



height, and an area for the disk drives is accessible for loading diskettes.

The desk has a work top area of 27″ by 48″ and is built to a height of 30″. Models are also available for the Atari and TRS-80. $400.

Furnwood Manufacturing Inc., 5665 S.W. Carman Dr., Lake Oswego, OR 97034. (503) 636-1991.

CIRCLE 340 ON READER SERVICE CARD

HOME CONTROL FOR TRS-80

The Micro Commander is a software driven interface between the TRS-80 and the BSR X-10 system. It provides remote control for up to 256 lights and appliances in the home or office.

Because the Micro Commander is a direct interface to the AC power line, the user need not purchase the BSR Command console.

The 14K TRS-80 Basic program listing and 8080/Z-80 assembly language listing are included in the manual. $59.95.

Interface Technology, P.O. Box 383, Des Plaines, IL 60018. (312) 297-2265.

CIRCLE 341 ON READER SERVICE CARD

HEURISTICS OFFERS EXPANDED VOCABULARY

An expanded vocabulary option has been announced by Heuristics, Inc. The option makes available a 128-word vocabulary with either the Heuristics 7000 industry-compatible unit or the 5000 Lear Siegler plug-in unit.

Each "word" can be a phrase up to three seconds in length, and the unit can be trained or re-trained to accept the voice or voices of the users. $200.

The 7000, which interfaces with all RS-232C terminals, sells for $3,000. The 5000, which fits direclty inside Lear Siegler's ADM-3A terminal, sells for $2,000.

Heuristics, Inc., 1285 Hammerwood Ave., Sunnyvale, CA 94086. (408) 734-8532.

CIRCLE 342 ON READER SERVICE CARD

DATA ACQUISITION CARD FOR APPLE

Mountain Computer Inc. announces an A/D + D/A card for Apple II computers. The card is intended for applications in data acquisition and control. All functions are accomplished on one printed circuit card which occupies one peripheral slot in the Apple II computer.

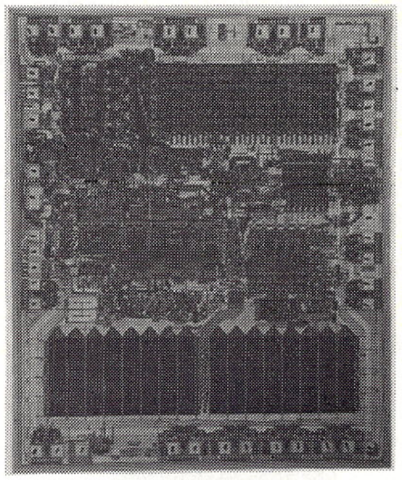
Its capabilities feature 16 channels analog to digital input, 16 channels digital to analog output.

It affords the Apple user the capabilities of data acquisition and control on one PC card, and with conversion speed and accuracy which permits high frequency applications.

Mountain Computer Inc., 300 Harvey West Blvd., Santa Cruz, CA 95060. (408) 429-8600.

CIRCLE 343 ON READER SERVICE CARD

COMPUTER PUZZLE



Micro-Madness is an actual microcomputer chip enlarged 80 times to become a 10" x 14", 252-piece jigsaw puzzle. The price includes the puzzle plus a full-color lithograph that is suitable for framing. $10.

Eubanks Engineering, P.O. Box 127, Valencia, PA 16059.

LOWER CASE FOR APPLE

Lazer Systems announces the Lower Case +Plus for the Apple II and Apple II Plus.

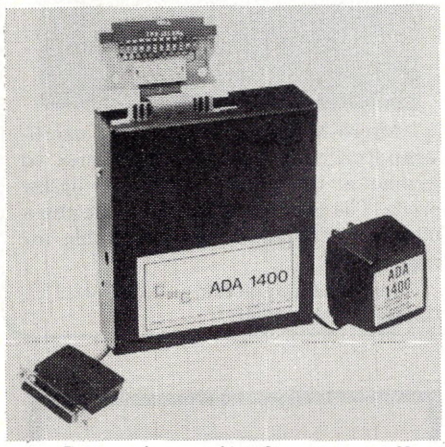
The plug-in board features two complete character sets on board. The primary character set is a word processing set that will place the two-character control function on the screen. The second character set is a Hi-RES graphics set that has pre-defined shapes such as cars, space ships and abstract graphics.

The expansion socket allows the use of an external RAM-based, user-defined character set. It also has Reset key disable capability. $59.95.

Lazer Systems, P.O. Box 55518, Riverside, CA 92517.

CIRCLE 344 ON READER SERVICE CARD

PET PRINTER ADAPTER



Connecticut microComputer offers an addressable printer adapter for use with all popular Commodore PET microcomputer models.

The ADA 1400 printer adapter drives any compatible microcomputer printer with RS-232 interface from the PET's IEEE-488 bus. The addressable ADA 1400 works with the Commodore disk and prints upper and lower case ASCII characters.

A PET IEEE type port is provided for daisy-chaining other devices such as PET disk, mainframe computer interface and IEEE-488 devices. $179.

Connecticut microComputer, Inc., 34 Del Mar Dr., Brookfield, CT 06804.

CIRCLE 345 ON READER SERVICE CARD

CUSTOMIZED TYPEHEADS AND PRINTWHEELS

Typing and word processing of technical materials can be simplified with the introduction of customized typeheads and printwheels from Dramco Sales, Inc.

Dramco replaces unwanted standard characters with symbols used in chemistry, engineering, mathematics, data processing and other disciplines. Company logos and custom ideograms can also be inserted into any typehead and metal or plastic printwheels which may be supplied by Dramco or the purchaser.

Dramco Sales, Inc., Panta, Suite 700, 620 Fifth Ave., NY 10020. (212) 489-2260, (800) 223-0979.

CIRCLE 346 ON READER SERVICE CARD

DATA COMMUNICATIONS FOR TRS-80

The Micromint Inc. announces the Chatterbox, a combination of the presently available COMM-80 I70 interface for the TRS-80 and an acoustic modem.

It includes a built-in programmable 50-19200 baud serial port, a Centronics compatible parallel printer port, a 300 baud acoustic originate modem, and a spare TRS-BUS expansion connector. It comes complete with power supply, connection cable, user's manual, and smart terminal software.

The Chatterbox is the only peripheral needed to allow a TRS-80 to communicate with timesharing systems such as Micronet and the Source. $259.95.

The Micromint Inc., 917 Midway, Woodmere, NY 10098. (516) 374-6793.

CIRCLE 347 ON READER SERVICE CARD

RESERVE POWER SUPPLY FOR APPLE

The Applejuice APS-3 is a reserve power supply for the Apple II. It supplies one hour back-up power to the Apple II and to Apple-powered peripherals.

During a power outage, the Applejuice alerts the user visually, audibly and electronically that there is a power failure, thus giving him time to bring his system to an orderly shut-down with all data files intact. $295. A 20 minute unit, the APS-2 is available for $249.

High Technology, Inc., P.O. Box 14665, Oklahoma City, OK 73113. (405) 840-9900.

CIRCLE 348 ON READER SERVICE CARD

PHOTOTYPE FROM MAGNETIC FILES

Resource Graphics announces a service that will produce high quality phototype output from magnetic files. This capability is offered to authors, editors, and publishers of small magazines, newsletters, brochures, proposals, reports, and books.

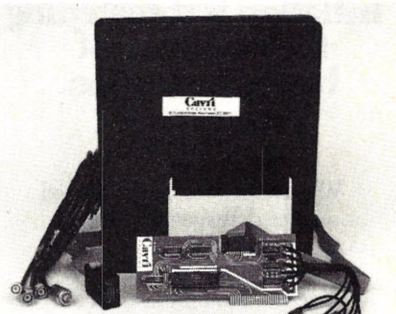
The system reads single density, 8″, soft sectored, CP/M compatible (or IBM compatible) diskette files, processing text which is in standard ASCII characters. Telephone linkage is also possible.

The phototype output runs 1 to 24 inches long by 8 inches in width, and can be set in over thirty different styles and in many sizes and formats. Rates begin at $.45 per 14 pica column inch for straight text matter.

William Burns, Resource Graphics, 7291 Coronado Dr., San Jose, CA 95129.

CIRCLE 349 ON READER SERVICE CARD

INTERACTIVE VIDEO



The Cavri III computer/video player integrator enables a user, seated at the computer keyboard, to index and later access a series of videotape frames or segments or to interact with videotaped materials. It also allows a user to control all remote functions of the VCR from the computer keyboard or from within a program.

The Cavri III System consists of an Apple I/O board, cables and connectors, system software in Applesoft Basic on disk, and a user's manual. It is available for VCRs that carry a control pulse or that interface with manufacturers' search units.

Cavri Systems Inc., 26 Trumbull St., New Haven, CT 06511. (203) 562-9873.

CIRCLE 351 ON READER SERVICE CARD

16K MEMORY EXPANSION KIT FOR YOUR TRS-80, APPLE, AND S-100 COMPUTER

only $59

- 200 Nsec Access, 375 Nsec Cycle
- Burned-in and Fully Tested
- 1 yr. Parts Replacement Guarantee
- Qty. Discounts Available

BETA COMPUTER DEVICES
1230 W. COLLINS AVE.
ORANGE, CA 92668
(714) 633-7280

EDUCATIONAL SOFTWARE TRS-80* 8K PET

80 + Programs In:

ELEMENTARY	MATH
SCIENCE	BIOLOGY
GEOGRAPHY	HISTORY
ECONOMICS	ACCOUNTING
FOREIGN LANG.	BUSINESS ED.
GAMES	FARM RECORDS

Programs are grouped into packages of 4 to 7 programs priced at $24.95 per package including shipping and handling. Available on disk or tape.

Write for catalog:
MICRO LEARNINGWARE BOX 2134
N. MANKATO MN 56001, 507-625-2205
Visa & MasterCard Accepted

* "TRS-80 is a registered trademark of TANDY CORP."

CIRCLE 212 ON READER SERVICE CARD

AT LAST!

Mass production prices on this high-quality software. Buy direct and save 50%. Now, also available for CBASIC on CP M and MBASIC on HEATH HDOS.

DATA BASE MANAGER Mod-I $69 Mod-II $199
You can use it to maintain a data base & produce reports without any user programming. Define file parameters & report formats on-line. Key random access, fast multi-key sort, field arith., label, audit log. No time-consuming overlays. 500 happy users in a year.

A/R Mod-I $69 Mod-II $149
Invoices, statements, aging, sales analysis, credit checking, form input, order entry. As opposed to most other A/R, ours can be used by doctors, store managers, etc.

WORD PROCESSOR Mod-I $49 Mod-II $49
Center, justification, indentation, page numbering. Mod-I version features upper/lower case without hardware change!

MAILING LIST Mod-I $59 Mod-II $99
The best! Compare and be selective. Form input, 5-digit selection code, zip code ext., sort any field, multiple labels. Who else offers a report writer?

INVENTORY Mod-I $99 Mod-II $149
Fast, key random access. Reports include order info, performance summary, E.O.Q., and user-specified reports. Many have converted their inventory system to ours!

GL, A/R, A/P, & PAYROLL Mod-II $129 each
Integrated accounting package. ISAM, 100+ page manual. Uses 80 column screen, not 64. A $1,000 value. Dual disk required.

L216, a cassette package of 10 business programs for Level II 16K systems, $59. Includes word processor & data base. Poker game $19.

MICRO ARCHITECT, INC.,
96 Dothan St., Arlington, MA 02174

CIRCLE 155 ON READER SERVICE CARD

WE WILL TRY TO SELL THE following product at the lowest ADVERTISIED PRICES IN THIS MAGAZINE

PET APPLE ATARI CROMEMCO

MISSISSIPPI MICROS, INC.
Mart 51, Jackson, MS. 39204
(601) 948-7846

CIRCLE 225 ON READER SERVICE CARD

ATARI 800 SOFTWARE!

ATARI 3-DIMENSIONAL GRAPHICS PACKAGE
HI-RES MULTICOLOR GRAPHICS TUTORIAL
8K to 48K....ONLY- $29.95 +$1.50 p&h

WUMPUS ADVENTURE - Graphics & sound
dynamic game...16K $14.95 +$1.50 p&h

DOWN THE TRENCH..Fast graphics, sound
uses 1 joystick.....8K or 16K..$14.95

BATTLING TANKS...requires 2 Joysticks
Game 8K.........ONLY- $9.95 +$.75 p&h

3-D RED-BARON DOGFIGHT/FLIGHT SYM.16K
uses 1 joystick......$15.95 +$.75 p&h

SUBMARINE MINEFIELD navigator......8K
uses 1 or 2 joysticks.$9.95 +$.75 p&h

BASIC EDITOR..8K-up...$8.95 +$.75 p&h

HARDWARE: DIRECT SOUND OUTPUT CABLE
W/SOFTWARE.....only $17.95 +$1.50 p&h

ON DISK- ADD $3.00 EXTRA PER ORDER
Calif. Residents Please add Sales Tax
SEND FOR YOUR FREE CATALOG TODAY!!!
FROM:
SEBREE'S COMPUTING....DEPT. 4S.
456 GRANITE AVE. MONROVIA, CA 91016
213-359-8092
FULL LINE OF 'BALLY ARCADE' SOFTWARE!

CIRCLE 223 ON READER SERVICE CARD

EDUCATORS... Are You Using Microcomputers?

A major publishing company is seeking reviewers of CAI Software for grade levels K-12.

Reviewers should have experience with classroom use of one or more microcomputing systems (PET, Apple, TRS-80, etc.).

Write:
Dept. A
900 Sylvan Avenue
Englewood Cliffs, New Jersey
07632

CIRCLE 235 ON READER SERVICE CARD

NEW MUSIC SOFTWARE

TRS-80 LEVEL II (16K)

MINI-KEYS play 7 octaves like any keybd. instrument. 100's of speeds. **$12.95**

ReKord — hear prerecorded data play. Stars & Stripes Forever, Hallelujah! **$16.95**

MYTEE MUSIC — assemble over 3000 tones in a music score. E-Z to use. E-Z to edit. Whole notes thru 1/64, dotted & triplet. 100's of speed signatures. Instruction booklet makes program fun to use!

MYTEE MUSIC-1	good	**$14.95**
MYTEE MUSIC'2	better	**$29.95**

MYTEE MUSIC-1	good	**$14.95**
MYTEE MUSIC-2	better	**$29.95**
MYTEE MUSIC-3	best	**$44.95**

(add $5.00 for disc)

MYTEE MUSIC
P.O. Box 2432
Evansville, IN 47714

CIRCLE 227 ON READER SERVICE CARD

TAXMAN

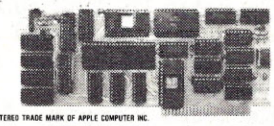
Comprehensive computerized tax system for O.S.I. Developed and implemented since 1972. System proven by many thousands of tax returns having been processed. Over 25 federal schedules, some state schedules. System compares to *any* system on the market and is *better* than most. System is 75 to 80 percent less expensive per return than other systems. Returns can be computed while client waits, or overnight. Excellent for C.P.A.s, lawyers, tax services, bookkeeping services or anyone who does tax returns.

Call TAXMAN at
(801) 485-2122 or (801) 487-9292
Or write Taxman,
1566 South Main, SLC, Utah 84115

CIRCLE 240 ON READER SERVICE CARD

80X24 VIDEOTERM™

7X9 MATRIX DISPLAY FOR
LOWER CASE W/ DESCENDERS **APPLE II®**



® REGISTERED TRADE MARK OF APPLE COMPUTER INC.

80 columns by 24 lines with easy to read 7x9 dot matrix, upper and lower case with descenders using shift lock feature • 1K firmware incorporates PASCAL and BASIC protocalls so user is not required to enter machine language programs or change PASCALS. Misc. info. or Gotoxy files • Compatible with all APPLE II peripherals so user won't need new software patches for future software products • Crystal controlled dot clock for excellent character stability • VIDEOTERM is the same size as the Apple language card and power consumption is held to a minimum through the use of CMOS and lower power devices • Character set can be user definable up to a maximum of 128 symbols of 8x16 dot matrix font • Display control character mode and four standard display formats controlled by escape sequences • Built in light pen capability • Inverse display mode • 50/60 HZ operation • Sockets on all IC's.

PRICE: $345
OPTIONS: Character Sets $39
VIDEO SWITCH PLATE, inserts in case slot to choose between APPLE II® and VIDEOTERM $19
MANUAL: $19

VISA

VIDEX 897 N.W. Grant Ave., Corvallis, OR 97330 Phone: (503) 758-0521

CIRCLE 245 ON READER SERVICE CARD

TRS-80 DIRECT CONNECT MODEM

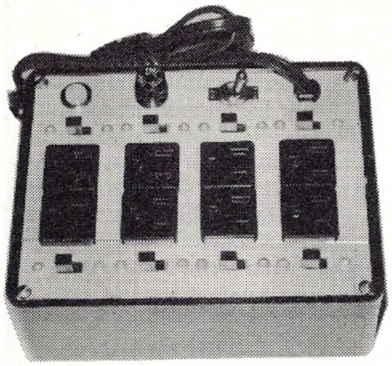
The MicroPeripheral Corp. announces The Micro Connection, an integrated RS-232 adapter and direct connect telephone interface. It is designed to be used with any model TRS-80 and connects to the computer data buss. It eliminates the need for the Radio Shack expansion box, RS-232 adapter and acoustical telephone interface in telecommunications applications.

The Micro Connection can adapt any TRS-80 for telecommunications with the Source, MicroNet, public bulletin boards, or virtually any computer system operating at 300 baud, $249.

The MicroPheripheral Corp., P.O. Box 529, Mercer Is., WA 98040. (206) 454-3303.

CIRCLE 352 ON READER SERVICE CARD

POWER CONTROL CONSOLE



Kalglo Electronics Co., Inc. has introduced a computer power control console, the Spike-Spiker. The compact desktop console allows the user to plug all computer equipment into one unit and switch the equipment on and off in the required order.

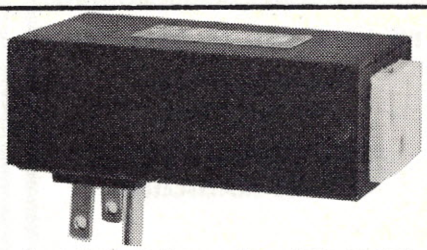
The Spike-Spiker also protects the computer from power line transients with an absorber and provides RF "hash" filtering between computer and motorized equipment in the computer system, home or office. The console has 8 individually switched 120 VAC outlets divided into two separate filtered circuits, main on/off switch, fuse and indicator light. $44.95.

Kalglo Electronics Company, Inc., Colony Drive Industrial Park, Box 2062, Bethlehem, PA 18001.

CIRCLE 353 ON READER SERVICE CARD

POWER-LINE FILTER

Television interference from your home computer and peripherals can be controlled by the EMI-117, 600-watt power-line filter from Marine Technology.



Installed at the wall socket used by the computer, it provides up to 60 dB attenuation at TV/FM frequencies, preventing interference from reaching the home power wiring and being conducted/radiated to television receivers in the vicinity. The filter is useful with noisy home appliances as well. $24.95

Marine Technology, 2730 Temple Ave, Long Beach, CA 90806. (213) 595-6521

CIRCLE 354 ON READER SERVICE CARD

SUPERBOARD II MOD. KIT

The Super-Mod Kit for the C1P and Superboard II provides a 48 character/26 line video display and software selection of 300 or 1200 baud for cassette and RS-232 operation.

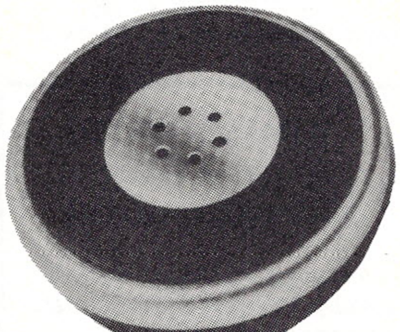
The kit also provides an RS-232 Port; start/stop control of the cassette; and doubling of system clock speed (from 1 MHz to 2 MHz). Instructions are included to add voice cueing and listening function during data transfer to and from cassette. The OSI Monitor PROM is replaced by a expanded Monitor PROM to include the above functions while still allowing the computer to be "booted up" in the normal manner.

Installation time is about 12 hours. $95.

A.H. Systems Inc., 9710 Cozycroft Ave., Chatsworth CA 91311. (213) 998-0223.

CIRCLE 355 ON READER SERVICE CARD

MODEM MICROPHONE



A modem microphone designed to give pure, clear data transmission has been introduced by Novation, Inc.

Engineered specifically to eliminate data distorting second harmonics, the FCC registered Super Mike slips into the telephone handset replacing the existing carbon microphone. $9.95

Novation, 18664 Oxnard St., Tarzana, CA 91356.

CIRCLE 356 ON READER SERVICE CARD

WIRELESS KEYBOARD INPUT FOR TRS-80

Omni Automation announces a wireless keyboard for the TRS-80. The model RX-10 includes a hand held, ultrasonic remote which can be used across the room.

The system also permits control of remote devices in the home or office. Communication to the remotes is via the A.C. power lines. A flexible scheduling program can activate the remotes automatically using cyclic, time of day, or future date schedules.

The RX-10 system includes all necessary cables, interfaces, cordless controller, command console, appliance and lamp control modules. Software is provided on diskette for status display, security monitoring, and scheduling.

Omni Automation, P.O. Box 7716, Atlanta, GA 30357.

CIRCLE 357 ON READER SERVICE CARD

CRYPTOGRAPHY KIT

Western Digital Corporation announces the availability of a Crytographic Primer Kit. The kit educates home computer users about cryptography, and enables them to encode and protect their own data against unauthorized access.

Each kit includes: an RS232 Interface Cryptographic Board in either kit or preassembled form which attaches to the RS-232 port of a computer; a "Cryptographic Primer" with several different cryptographic architectures using the Data Encryption Standard; "Codebreakers" by David Kahn which gives the reader a perspective of cryptography's impact on history and an assembly/wiring manual which includes wiring diagrams, assembly instructions and operating instructions. Kit, $395; assembled, $495.

Western Digital 3128 Redhill Ave., Newport Beach, CA 92663. (714) 557-3550.

CIRCLE 358 ON READER SERVICE CARD

CALIFORNIA

Rainbow Computing, Inc.—9719 Reseda Blvd., Northridge 91324; (213) 349-5560. 10-7 Tues-Fri, 10-5 Sat, 12-5 Sun. Apple, DEC, and Atari. Authorized Sales and Service.

D.E.S. Data Equipment Supply—8315 Firestone, Downey 90241. (213) 923-9361. 7 days. Commodore PET specialists. Hardware Software, Books, Mags, Supplies, In House Maintenance.

Advance Data Concepts—2280 Diamond Blvd., Concord, 94520; (415) 671-9016. 9-5 Mon-Fri. Vector-Graphic, CP/M Software Headquarters-User's Group.

CONNECTICUT

Computerworks—1439 Post Rd., East Westport 06880; (203) 255-9096. 12-6 Tues-Fri, 12-9 Thu, 10-5 Sat.

FLORIDA

AMF Electronics—11158 N. 30th St, Tampa 33612; (813) 971-4072. 10-6 Mon-Sat. Apple Computer Sales & Service; TRS-80, Apple Software & Peripherals; S-100 boards, computer parts & books.

GEORGIA

Atlanta Computer Mart—5091 Buford Hwy, Atlanta 30340; (404) 455-0647. 10-6 Mon-Sat.

ILLINOIS

Computer Junction—543 S. York Rd., Elmhurst 60126; (312) 530-1125. Mon & Thu 9:30-8:30 pm; Tues-Sat 9:30-5:30; Sun 12-4:30.

The Computer Room—106 E. Oak St, Chicago 60611; (312) 337-6744. 11-7 Mon-Fri, 11-6 Sat.

Data Domain of Schaumburg—1612 E. Algonquin Rd, Schaumburg 60195; (312) 397-8700. 12-9 Tue-Fri, 11-5 Sat. Largest book & magazine selection.

Lillipute Computer Mart, Inc.—4446 Oakton, Skokie 60076; (312) 674-1383. M-F 10:30-8 pm, Sat 10-6. We sell Cromemco, Gimix, Bell & Howell, NorthStar and others. Starting our fifth year in business.

KENTUCKY

ComputerLand of Louisville—10414 Shelbyville Rd, Louisville 40223; (502) 245-8288. 10-5:30.

MASSACHUSETTS

Computer Mart, Inc.—1395 Main St, Waltham 02154; (617) 899-4540. 11-6 Tue-Sat. Atari, Heath, NEC, SWTP & S-100 bus systems; Word Processing Specialists.

Neeco—679 Highland Ave, Needham 02194; (617) 449-1760. 9-5:30 Mon-Fri. Commodore, Apple, Superbrain, TI 99/4.

NEW JERSEY

Computernook—Rt. 46, Pine Brook Plaza, Pine Brook 07058; (201) 575-9468. 10-6:30 MTWS, 10-8 Thurs., Fri. Apple/Commodore Authorized dealer.

Stonehenge Computer Shop—89 Summit Avenue, Summit 07901; (201) 277-1020. 10 am - 6:30 pm Mon-Sat. Apple/Bell & Howell/Commodore Authorized Dealer, Sales and Service.

NEW YORK

The Computer Corner Inc—200 Hamilton Ave, White Plains 10601; (914)-WHY DATA. 10-6 Mon-Sat, 10-9 Thu.

OHIO

The Basic Computer Shop—2671 W. Market St, Akron 44313; (216) 867-0808. 10-6 Mon-Sat.

Micro Mini Computer World—74 Robinwood Ave., Columbus 43213; (614) 235-5813/6058. 11-7 Tues-Sat. Authorized Apple/Commodore dealer. Sales, Service, Business Software.

PENNSYLVANIA

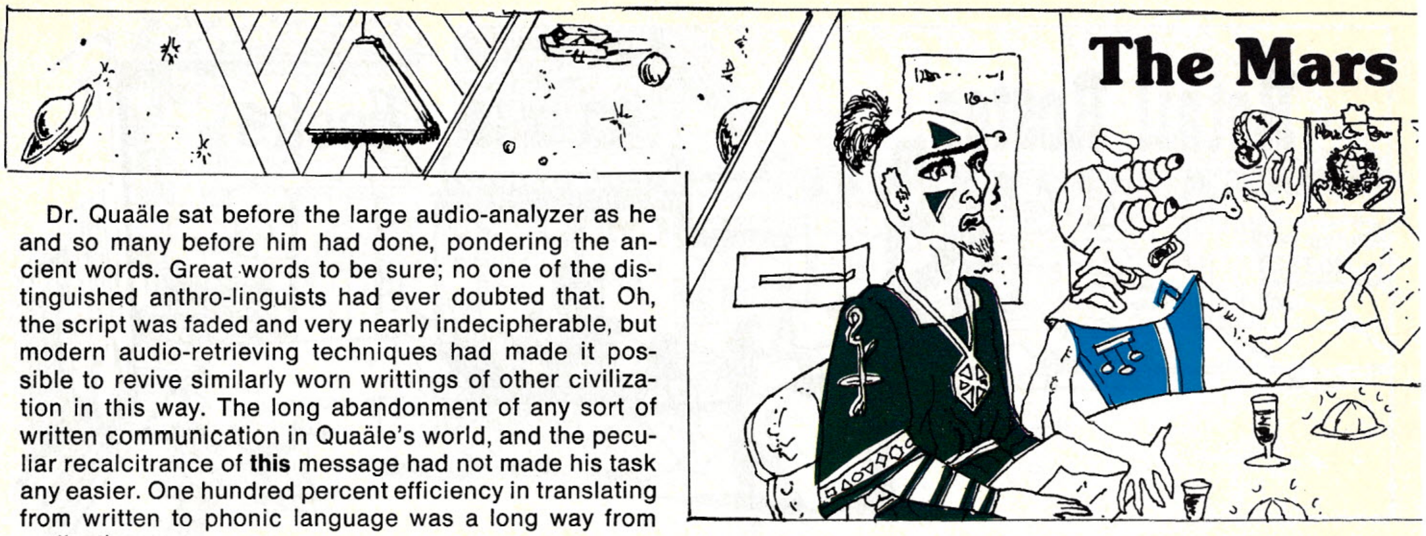
Personal Computer Corp.—24-26 W. Lancaster Ave, Paoli 19301; (215)647-8463. 10-6 Mon-Fri, 10-8 Wed, 10-5 Sat.

VIRGINIA

ComputerLand/Tysons Corner — 8411 Old Corthouse Rd, Vienna 22180; (703) 893-0424. 10-6 MTWF, 10-9 Thu, 10-5 Sat.

Computer Plus, Inc—6120 Franconia Rd, Alexandria 22301; (703) 971-1996. 10-9 Mon-Fri, 10-6 Sat. Micro specialists, books, classes, software, maintenance. "The PLUS makes the difference."

The Mars



Dr. Quaäle sat before the large audio-analyzer as he and so many before him had done, pondering the ancient words. Great words to be sure; no one of the distinguished anthro-linguists had ever doubted that. Oh, the script was faded and very nearly indecipherable, but modern audio-retrieving techniques had made it possible to revive similarly worn writtings of other civilization in this way. The long abandonment of any sort of written communication in Quaäle's world, and the peculiar recalcitrance of **this** message had not made his task any easier. One hundred percent efficiency in translating from written to phonic language was a long way from realization.

But the upcoming meeting was to be the summit, so far as the words on the ancient container were concerned. All the great scientists, linguists, social scientists and theologians of Merador had been invited. Either a definitive analysis would be forthcoming, or the writings would be fed again into the memory banks for another millenium and another summit. Quaäle was hopeful this time; he had, after all, the very latest techniques

He had the very latest techniques of linguistic analysis available to him.

of linguistic analysis available to him. His task was clear. First the usual breakdown into the phonetic, phonic, phatic and locutionary levels of speech. This had not been difficult, although some extrapolation from Meradorian syntatic structures to the Ancient syntax had been necessary. More problematic would be the subsequent analysis into the elocutionary and rhetorical levels. Hopefully this would give some insight into the semantic, as well as the syntactic deep structure of the script. Doubt gnawed at him nevertheless. He recalled the great clash early in the 22nd century between the linguists and the philosophers of language on the great questions of literal vs. symbolic meaning and denotation vs. connotation. Would the usual analysis into noun phrases and verb phrases alone suffice? What of the vexing question of the levels of sense? What of the interpretive, allegorical and metaphorical levels of meaning?

"Well," he mused, "can't worry about **that** right now! I'm sure the philosophers and theologians will have a ball with it anyway!"

Aware of the impending summit and the imminent entrance of its participants, Quaäle switched on the audio-enlarger for the last time. The machine delivered its measured monotone:

"Mare-akareist-muss-frum-be-ing
Mars-acreryst-mass-frumbin
Mar-sacral-lyte-massfrumbing
Mars-akareist-massfrum-binn
Mer-okar-rytemuss-frumbin
Mars-acre-heist-mess-frumbin
Mar-okar-heist-mass-frum-be-ing
Mare-ache-her-hite-miss-frumbin
Mars-ache-her-heistmussfrum-bin"

Anthony Serafini, Assoc. Prof. of Philosophy, Boston State College, 625 Huntington Ave., Boston, MA 02115.

Nothing.

"Not a recognizable verb in the whole mess," he noted disgustedly.

Bits and pieces to be sure, but hopes of finding some kind of coherent sentence structure seemed to be fading. No time for further private speculations now, as the conference was about to begin. Dr. Quaäle greeted each of the distinguished panelists in turn and summarized progress to date:

"You will notice, gentlemen, that efforts to analyze the ancient words into their deep phrase structure have so far failed. A number of different analyses offer themselves as candidates for nouns, noun phrases, articles and the like, but a clear sentential structure has yet to emerge."

"Excuse me, Dr. Quaäle," interrupted Cónraynëran, the distinguished sociologist, "but isn't it just possible that the symbols have no sentential structure? Could it be a name — perhaps of some great personage in the history of that civilization, for example?"

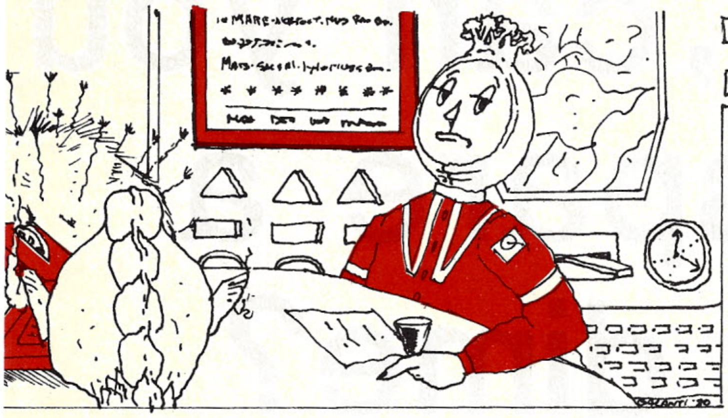
"Certainly that's a possibility," responded Quaäle. "You will notice that the name of the planet Mars occurs fairly frequently in the computer's speakouts. It has been suggested by the astronomer Zaekbar that the whole phrase refers to some sort of cataclysmic event on that planet. If it please the participants, I will ask Prof. Zaekbar to elaborate."

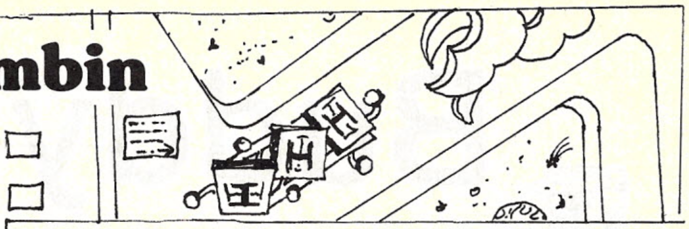
The eminent astronomer and astrophysicist crawled swiftly to the analysis screen and began his presentation.

"Quite obviously, these words refer to some sort of physical catastrophe of immense proportions on Mars. It would seem that there are two reasons for thinking so. For one thing, the frequent occurrence of the term 'mass' in the readouts. Now, we know that in the approximate time period under discussion, there was great volcanic activity on Mars with subsequent shifts of great land masses. For another thing, we can see that the entire phrase actually occurs several times on the container in question. We cannot say just how many times, of course, due to the poor condition of the relics. Nevertheless, the sheer number of repetitions suggest that they are associated with an inordinate degree of busy, almost frantic activity. Most likely a holocaust of some kind."

Zaekbar's remarks were greeted by general nodding and approval, though a number of the theoreticians present voiced objection to this analysis.

Acre-Heist Mess Frumbin





Anthony Serafini

"Excuse me," chimed in the biologist Jola, "Surely we cannot overlook the frequency of biological terms in the speakouts. Notice that symbols like 'ache,' 'sacral' and 'okar' have very definite biological connotations. We must remember, gentlemen, that the 22nd century was a time of great biological as well as geological upheavel throughout the galaxy. I am willing to accept, as at least a working hypothesis, Quaäle's and Zaekbar's suggestion that the planet Mars figures importantly in our understanding of the words. But it seems to me that they refer to some sort of great biological catastrophe — perhaps a sudden evolutionary shift. Though our records are imprecise on this question, there is good reason to believe that the Okar seed wreaked havoc in some way with the planet's experimentation in breeding species across dimensional lines. Most of the planet's faunae and florae either died or took ill. To be sure, the major portion of the trouble occurred late in the 21st century, but a mutated species of Okra may have arisen later, i.e., in the period in question."

"Gentlemen, gentlemen," pleaded Dr. Quaäle, striving to quiet the agitated group of scholars. "We have let the scientific community have their say; now in all fairness we must hear from the most distinguished historian Prof. Wombátor."

"Thank you Dr. Quaäle, I do in fact have a theory on the writings. First off, let me add my voice to those who see the words as referring to Martian phenomena; I also am of one mind with Dr. Quaäle — on that point anyway. Now, my intention is not to upstage Dr. Quaäle on linguistic niceties, but I believe we all have overlooked possible slang and regionalistic interpretations of the words. Please to note that the term 'heist' occurs in the computer speakouts, a slang, but meaningful, expression which fell into disuse sometime around the beginning of the 21st century on Mars, though it disappeared from the vocabulary considerably earlier on Earth and other planets. I believe you will find that it is synonymous with 'steal,' 'rob' or something of that order. 'Acre,' of course, was and still is, a term denoting a segment of land coverage in many parts of the galaxy. Recent studies in the **Tapes of Galactic History** have shown that the period of time we are talking about was one of violent, and frequently bloody, social change, often coincident with geological and biological upheavels spoken of earlier in this colloquium. The magnitudes of these changes was perhaps less awesome on Mars than on such notoriously savage worlds as Jupiter and Titan, but awesome nonetheless. The speakout I would prefer, therefore, is the **'Mars Acre-Heist Mess Frumbin.'** I should not be at all surprised if it refers to some cheap tabloid story on the great land swindles occurring on Mars during that period."

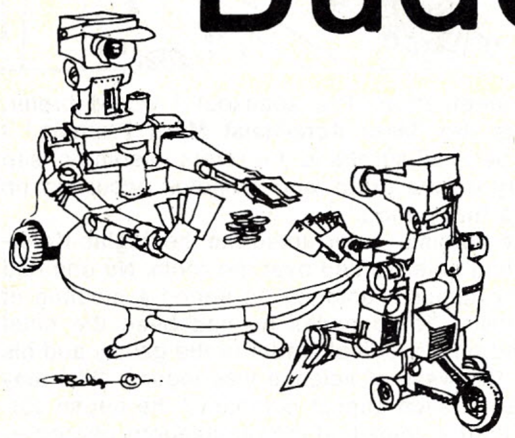
"Absolute rubbish," interrupted the theologian Kolërlan. A startling quiet settled over the room. No one had expected the famed theologian to attend a meeting of mere scientists. For decades he had been the chief religious and philosophical leader in the galaxy and his disdain for the ways of science was legend. "If I may have my say in such august company," he began sardonically, "I rather think that the speculations offered so far are altogether too thin to build anything resembling a plausible interpretation. Bear in mind, gentlemen, I am not suggesting you discontinue your noble efforts, for whatever good it will do. You must now look to the real sciences of theology and philosophy, however, for **ultimate truth.** A growing number of my colleagues feel certain that the words are traces of what we would call "beings." Notice the final phonic units in each speakout. We feel these beings to have been ancient travellers, most likely divine, or very nearly so. Further, if one studies the symbols on the containers one can see the hand of the Divine at work, much as the great philosophers of antiquity claimed that the Purposeful nature of the Universe is revealed in His creations. That we possess only the container and not the contents is doubtless part of the Divine Plan which will one day be revealed to us. You scientists try to decipher the words, but the meaning eludes you as it always will. A true understanding can only come from within the depths of your consciousness. You must look beyond the literal; only then will you hope to begin to unravel the truly profound story encapsulated in the words. Only then will you be one with the divine intentions of the holy and all-powerful 'being' who sent us this message.

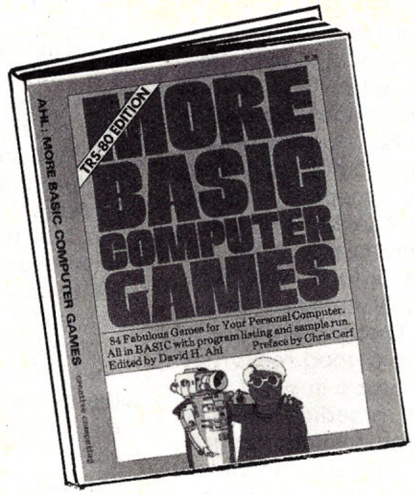
You must look beyond the literal; only then will you hope to begin to unravel the truly profound story encapsulated in the words.

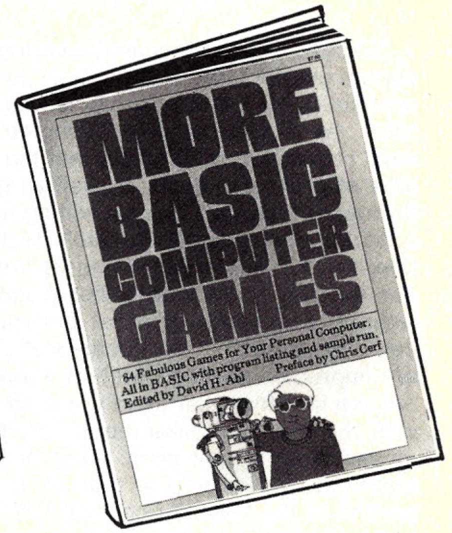
Quaäle was stunned; Kolërlan's presentation virtually guaranteed the failure of the colloquium, so great was his influence in the galaxy. The transcript of the proceedings would be published of course. More than likely though, the thoughts of Kolërian had slammed the door on any chance of open-minded discussion of other theories.

Following brief conversation with each of the participants, Quaäle retired to his study to feed the writings to the memory banks of the Central Computers, as so many of his predecessors had done. Deftly working over the micro-input switches, he located the mnemonic cell where the crumbling relic would be filed for the benefit of future scholars. Wondering if anyone would ever solve the mystery, he looked one more time at the ancient and still indecipherable communication: "Merry Christmas from Bing," and fed it into the computer. □

Buddy, can you spare a dime?







Contents

The programs in **More Basic Computer Games** and its new version **More Basic Computer Games: TRS-80 Edition** cost less than 10¢ each. The micro chip brought down the cost of hardware and Creative Computing brings down the cost of software.

This fantastic sequel to **Basic Computer Games** contains sample runs, program listings and descriptions for 84 new games.

Talk to Eliza, the psychologist, evade a man-eating rabbit, crack a safe, tame a wild horse, become a millionaire, race your Ferrari, joust with a Knight, trek across the desert on your camel, navigate in deep space, hunt a wumpus and much more.

Rather be on a space adventure? Play Close Encounters, Deepspace or ICBM.

In **More Basic Computer Games** all the games run in standard Microsoft Basic and a Basic Conversion table is included.

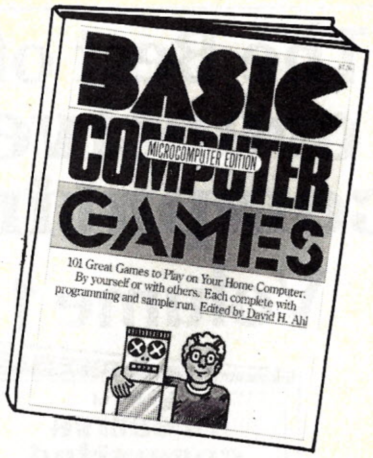
In the new **More Basic Computer Games: TRS-80 Edition**, all 84 games are converted to run on Level II 16K TRS-80 machines. Radio Shack users will delight that the conversion work on these imaginative and challenging games has been done.

Dr. Dobb's Journal of Computer Calisthenics & Orthodontia says "Whether you are interested in war games, gambling, sports, grids and mazes, space, psychology, drag racing or throwing mud balls, **More Basic Computer Games** has something in it for you."

Edited by David Ahl and Steve North, both books are a large 8½ x 11", Softbound, 200 pages. Each is $7.95. **More Basic Computer Games** (6C2). **More Basic Computer Games: TRS-80 Edition** (6C4).

To order, use bound-in card in back of magazine.

Or how about 8¢?



Why is Creative Computing's most popular book now in its fourth print run, having sold over one hundred thousand copies? Simple. Because it gives users the excitement they want from their computers.

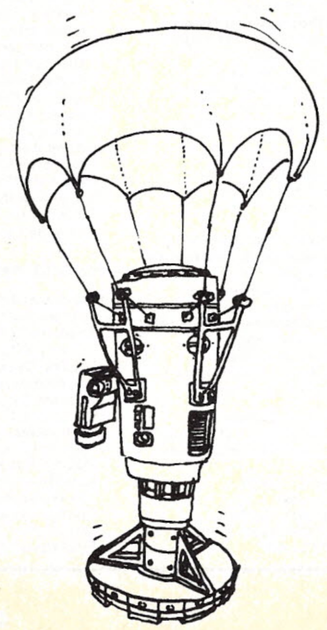
Basic Computer Games is a complete anthology of 101 favorite games and simulations, each complete with sample run, program listing, and description. All games run in standard Microsoft Basic and are easy to use with any computer.

Command the Enterprise against the fleet of Klingons menacing the United Federation of Planets. Learn to simulate a parachute jump successfully, not with a splat. Play High I-Q, Hi-Lo, or Mastermind and improve your powers of logic. On a wet winter day figure a way to manage the city of Hammurabi so your people don't starve. Let your computer print you a calendar for the year 1984.

There are games just for fun, to develop logic strategies, to teach environmental solutions, to simulate athletic competition, to play alone or with a group. The 101 games offer the kind of challenge, excitement, and delight you want from your computer, and your computer wants from you.

Edited by David Ahl and outrageously illustrated by George Beker. Large format paperbound, 200 pages, $7.95. (6C).

All games available on two 8" CP/M disks. $24.95 each.



Contents

creative computing press

To order, use bound-in card in back of magazine.

The Best of creative computing Volume 1

All the articles, stories, learning activities and games that appeared in the first year of Creative Computing. **American Vocational Journal** says "This book is the 'Whole Earth Catolog' of computers." Contents cover the gamut of computer applications in education and recreation. Over 200 contributors! Now in its fourth print run with an exciting new cover. 328 pp. 8½ x 11" Softbound. (6A) $8.95

Partial Listing of Contents -

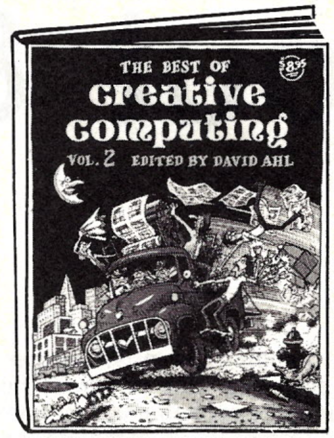
ARTICLES AND COMMENTARY

- **Editorials**
  Birth of a Magazine — Ahl
  A Computer in the Classroom?
  Is Breaking Into A Timesharing System A Crime? — Tagg
  Where Are We Going? — Ahl
- **Computers in Education**
  What's Wrong With the Little Red Schoolhouse? — Ahl
  How to Cope With Your Computer
  Recent Trends in Mathematics Curriculum Research — Critchfield
  CITALA: Computing in a Two-Year College — Howard, et al
  EXPER SIM: Experimental Simulation —
  Monty Python Meets Monte Cristo —
  IFIP Conference Report — Hebenstreit
- **Transportability**
  The Parable of the Horse — Nevison
  Technical Transport Problems —
  CONDUIT Documentation Guideline
  Statewide Pools May Not Yield Expected Benefits — Magarrell
- **Hard Core CAI**
  PLATO IV System Progress —
  TICCIT System Progress —
  PLANIT: The Portable CAI System
- **Careers**
  A Computer Career for You?
  Career Education: Will It Last?
  Key to Your Future? — Corr
  Profile of an Industry
- **Applications**
  Computers and the Weather
  Computer Simulation of the Atmosphere
  Weather Forecasting Applications
  Relativity for Computers: All Arithmetic
  Mr. Spock's 7th Sense — Kibler
- **Programming and Languages**
  Structured Programming — Hoogendyk
  On Computer Languages — Ahl
  Toward A Human Computer Language — Cannara
  Learning About Smalltalk — Goldeen
  Eclectic Programming Languages
  A New Approach to Testing —
- **Computer Impact on Society**
  The Computer Threat to Society — Ahl
  Digital Calculators — Then and Now
  The Computer. Threat to Society? —
  Putting Teeth Into Privacy Legislation — Hastings
  Industry Leaders at Privacy Hearings — Hastings
  Record-keeping in the Space Age —
  A Manufacturer Looks at Data Privacy — Fritze
  Survey of Public Attitudes Toward Computers — Ahl
  NBS Privacy Conference
  How Much Privacy Should You Have —
  Memoirs of an Ex-Social Security Number Giver — Campbell
  Crime, Cops, Computers — Malcolm
  Prosecutor Management Information System — Ahl
  A Computerized Criminal Justice System — Boekelman
  Embezzler's Guide to the Computer
  Credit Card Crooks
  Waiting for the Great Computer Rip-Off — Hastings
  Computer Abuse — Snyder

PEOPLE, PLACES, AND THINGS

Nicholas Copernicus
Evelyn Roth
PILOT 73 Information Exchange —
Nolan Bushnell — Todd
Playing PONG to Win — Ahl
Your Own Computer? — Ahl
Introducing Computer Recreations Corp. — Todd, Guthrey
Creative Computing Compendium
Flying Buffalo — Loomis
Compleat Computer Catalogue
National Computers in Education Conference?

The Best of creative computing Volume 2



A staggering diversity of information and activities culled from the issues of the 2nd year of Creative Computing. Includes feature on artificial intelligence, on computers in education, on the arts. **American Libraries** says "Non-technical in approach, its pages are filled with information, articles, games and activities. Fun layout." 67 pages devoted to puzzles, programs, and things to do. The reviews alone could make the book. 336 pp. 8½ x 11" Softbound. (6B) $8.95.

Partial Listing of Contents -

ARTICLES AND COMMENTARY

- **Technology — Present and Future**
  The Future of Computer Technology -
  Computing Power to the People
  Videodiscs — The Ultimate Computer Input Device? - *Bork*
  Round and Round They Go
  The $2.98 Computer Library -
  Personal Computers
  Russian Computing - *Ahl*
  Desk Calculator from China - *Chang*
  Microprocessors & Microcomputers The State of the Art - *Callahan*
- **Languages and Programming Theory**
  The Reactive Engine Paper - *Winograd*
  About Computing - *Chase*
  David vs. 12 Goliaths - *Newborn*
  Sixth Chess Championship Summary
  Beating the Game - *Thomsen*
  Simulated Strategies of Game Playing *Reisman*
  Beyond BASIC - *Salisbury*
  The Computer "Glass Box" Teaching with APL - *Peelle*
  Creative Chess - *Koetke*
  SNOBOL - *Touretzky*
  A Smalltalk Airplane Simulation - *Horn*
- **Artificial and Extraterrestrial Intelligence**
  Non-Human Intelligence - *Ahl*
  An Esoteric Ethical Excursion - *Lees*
  The Thinking Computer - *Raphael*
  Primer on Artificial Intelligence - *Garrett*
  Can Computers Think - *Ahl*
  An Ear on the Universe - *Lees*
  Communication Across the Universe
  The Cosmic Subway Line - *Asimov*
- **Literacy, Philosophy, Opinion**
  What is Computer Literacy - *Moursund*
  Computer Literacy Quiz - *Moursund*
  A Fable - *Spero*
  Let Us First Make It - *Taylor*
  Some Thoughts - *Lees*
  Information Anyone? - *Griffith*
  The Government Dinosaur - *Winn*
  The Magic of EFTS - *Ahl*
- **Computers in Education**
  Instructional Computing in Schools - *Ahl*
  Should the Computer Teach the Student, or Vice-versa? - *Luehrmann*
  The Art of Education: Blueprint for a Renaissance - *Dwyer*
  Computing at the University of Texas
  Computers in Secondary Schools - 1975
  Compyouter Fair - *Thomas*
  The Madness known as
- **Every Person and the Computer**
  Amateur Computing - *Libes*
  A Retail Computer Store? You Gotta Be Kidding - *Dunion and Roberts*
  Grand Opening - *Cary*
  Polls, Pols, and Power: The Computer on the Hustings - *Acocella*
  An Analytic Examination of Creative Computing - *Ahl*
  How We Spent Our Summer Vacation - *Lees, et al*
- **Art and Poetry**
  Toward the Electric Symbol - *Mueller*
  Producing Computer Poetry - *Chisman*
  Interview with Carole McCauley
  Once Upon a Computer
  Computers and Beauty

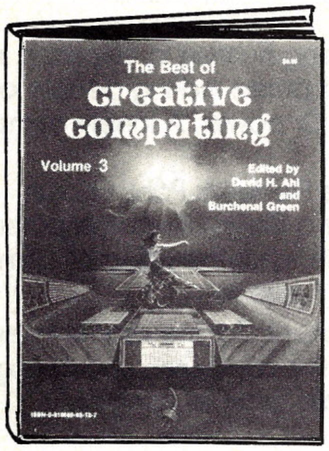
PUZZLES, PROBLEMS, AND PROGRAMS

- **Puzzles and Problems**
  Puzzles, Puzzles, Puzzles - *Ahl*
  Thinkers' Corners, Recreations
  Turning A Puzzle Into A Lesson - *Homer*
- **For The Calculator**
  The Keyboard Game - *Yarbrough*
  7 Pocket Calculator Games - *Rogers*
  Calculator Tricks
- **Mathematics and Geometry**
  The Mystic Seven - *Dickens*
  Magic Squares on the Computer - *Piele*
  Non-Usual Mathematics - *Reagan*
  The World of Series - *Reagan*
  Change For A Dollar - *Hess*
  Sequences - *Jessen*
  Progression Problems - *Reeves*

To order, use bound-in card in back of magazine.

The Best of creative computing

Hot off the press Volume 3



336 pages of articles, activities, fiction, games, programs, reviews, cartoons, and other information from the 1977 issues of Creative Computing. Includes features on technology, public access, educational use, medical applications, and computers in music. Contains great resource listings and reviews of calculators, games, equipment, software and books. There are 96 pages of things to do—puzzles, programs, problems, and games.

A sample of the diverse contents is listed.

Edited by David Ahl and Burchenal Green. Large format. 336 pages. $8.95 (12C).

Partial Listing of Contents -

•Technology—Present and Future
Trends Into the Future—Gray
EFTS: Living is Better Electronically, or IS it?—Dragunas
The World In Your Own Notebook—Lees
Eeny, Meeny, Micro and More—Salisbury
The Pocket Computer Is Almost Here—Ahern
Microprocessors—A Primer—Cohen

•Public Access
Computing at a Public Library—Shair
Computer Power to the People—Ahl
A Dream For Irving Snerd—Nelson
Time For a National Computer Club—Kuzmack
The Microcomputer Inflicts "Future Shock" on Technical Education—Vuillequiz

•Computers in Education
Interactive Computing in Secondary Schools in France-
A Microcomputer Software Course—Williams
Computer Science at Carnegie-Mellon University—Hast
Final Exams—Eisenberg
Computational Unsolvability—Steen
State-of-the-Art vs Compatibility—Ahl

•Languages and Programming Theory
Something Is Missing—Finseth
File Structures—Lees
PILOT—Yob
A Taste of APL—Finseth
ARTSPEAK—Friedman

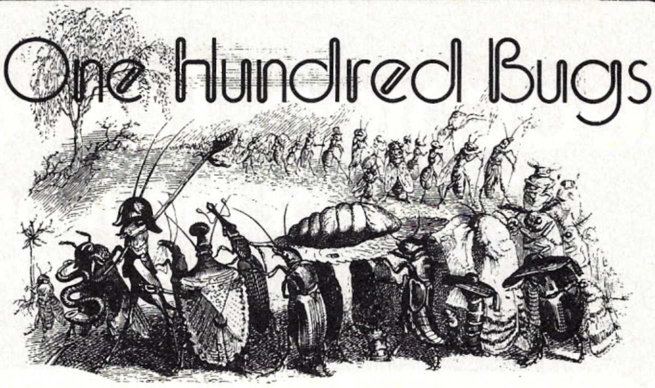
•Computers in Medicine and Science
A New Generation of Biomedical Instruments—Brus
The Miraculous Medical Microprocessor—Weintraub
Computerized Robots—Armstrong
Computer Correction of Optical Illusions—Smith
Brown Scientists Peer Into Fourth Dimension—Norris
An Inexpensive Reading Machine for the Blind—Brus
Medical Computerized Data Bases—Hastings
The Placebo and the Computer—Hastings

•Fiction and Foolishness
The Land of Halco—Rowlett
Them Hobbyists—Dunion
Computer Control—Vitale
Yellow Computer—Ragen
Edu-Man Meets Pseudo Hero—Ahl
Edu-Man Meets the Rumor Mongers—Ahl
The Lighter Side of Robots
The Lighter Side of Computer Dating
Nords—Sunstone Graphics
Glorobots—Maxson
The Floating Point Solution—Taylor
Martsport—Sonntag
Out of the Mouths of Babes—Wirth
Still a Few Bugs in the System

•Games
Othello—Wright
SWARMS—Miller
EUCHRE—Raybaud
Daytona 500—Churchill

•Reviews
Of Calculators
Sophisticated Electronic Pocket Calculators
Of Games
Smart Electronic Games—Ahl
Comp IV—Gray
Of Software
Software Technology Music System—Ahl
A Comparison of Software Systems—Ahl
Review of Five Small Interpreters—North
Notes on Languages—Chase
A Dynamic Debugging System—North
An Evaluation of Three 8080 8K Basics
New Benchmark Program—Chase
Two Space Games with Graphics

One Hundred Bugs



Engineers, programmers, managers, hobbyists, and other professionals need to write well. They have to be able to inform, persuade, and motivate their readers. Clear and concise writing is a necessary skill.

One Hundred Bugs helps computer people to write more effectively. It shows how to turn dull, difficult, awkward writing into lively, clear, clean writing.

After years of consulting with engineers, programmers, managers, and other technical professionals, Edmond H. Weiss reached two conclusions:

- Most writers make only a handful of errors, and
- Once the errors are pointed out, most writers can avoid them with ease.

One Hundred Bugs contains the errors Edmond Weiss has seen most often. As he puts it, "If your writing is free from these 100 kinds of mistakes, you're probably the clearest writer in your organization."

One Hundred Bugs is a "hands-on" book. Each of the 100 mistakes is contained in a problem sentence or passage. If the reader can spot the error and correct it, he can skip the material that refers to that particular "bug." If the reader can't figure out what's wrong, he reads the appropriate part of the book—and tests himself to be sure he's got it.

All the puzzles and examples in **One Hundred Bugs** are taken from real reports, proposals, and manuals.

The topics are: Words, In General...Verbs, In Particular...Phrases...Sentences...Links...Grammar... Punctuation and Mechanics..."Style."

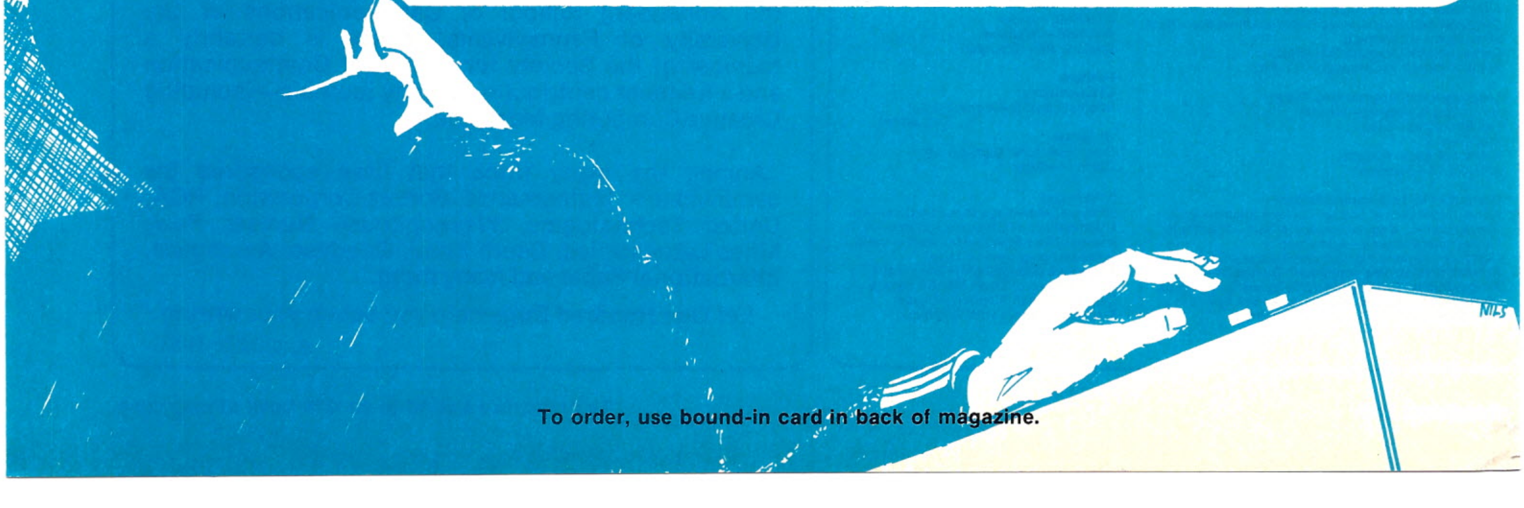
Dr. Edmond H. Weiss, President of Crown Point Communications, has conducted scores of communication seminars for thousands of technical professionals in four countries. Formerly Associate Dean of the Annenberg School of Communications of the University of Pennsylvania, Weiss is currently a member of the Society for Technical Communication and a frequent contributor to many journals—including Creative Computing Magazine.

Among the many firms that have sponsored his seminars are: Mathematics, Sorbus Corporation, RCA, United Technologies, Westinghouse Nuclear Fuel, Miles Laboratories, Smith Kline, Goodyear Aerospace, International Paper and many more.

Let **One Hundred Bugs** help you debug your writing.

Available 1981.

To order, use bound-in card in back of magazine.


NILS

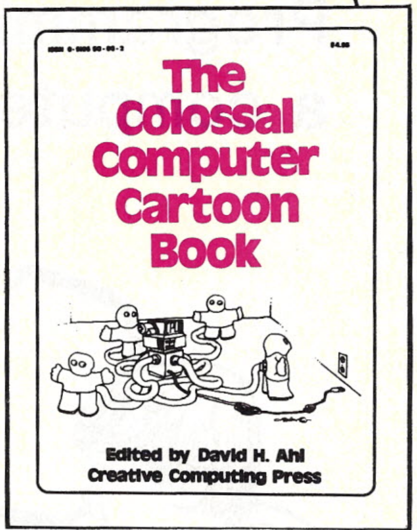
Do Computer Enthusiasts Have More Fun?

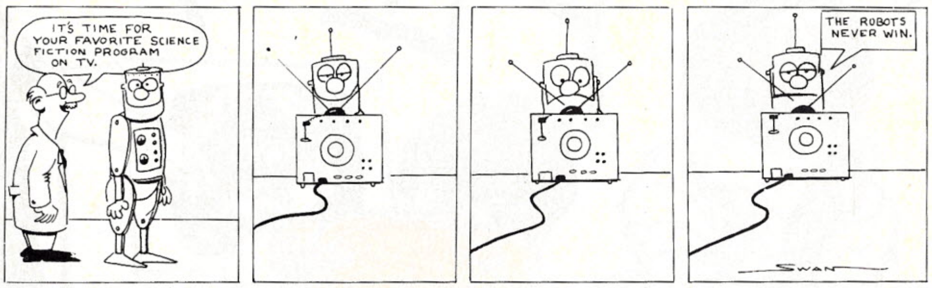
The Colossal Computer Cartoon Book

The best collection of computer cartoons ever is now in its second printing, and sports a bright new cover. The fifteen chapters contain hundreds of cartoons about robots, computer dating, computers in the office, home, and lab, and much more. 36 cartoonists share their views of man's ultimate machine.

Keep this book with your reference works. When needed, the right cartoon can say it all for you. When you need a break from debugging a good laugh can give you a welcome lift. Recommended for hours of fun and comic insight.

Edited by David Ahl, mastermind behind the April Fool's issue of Dr. Kilobyte's Creative Popular Personal Recreational Micro Computer Data Interface World Journal, this cartoon book contains much of that same incurable zaniness. [Want this issue? It's April 1980 and only $2.50 postpaid].





A large 8½ x 11" softbound collection of 120 pages, it still sells for only $4.95. (6G).

Bach Baroque Boogie by Binary

The 12" LP is one disk that's guaranteed to load every time, and guaranteed to bring pleasure. It provides the perfect background for programming, partying, or relaxing. $6.00 (CR101).

The Philadelphia Computer Music Festival has become a traditional favorite at the Philadelphia Computer Show. This live recording brings the first of those festivals into your home for an encore preformance. Eight different computer music systems play synthesized interpretations of J.S. Bach, J. Pachelbel, Rimsky-Korsakov, Scott Joplin, Neil Diamond, Lennon & McCartney, and seven others. To add a human touch, the record includes duets performed by computers and flesh-and-blood musicians.

The range of music, from baroque to rock, traditional to rag, will please anyone's taste. Also included is the historic 1963 computerized singing demonstration by Bell Labs.

Programming a computer is

Hey Kids, are the folks out of the room? Good, 'cause I've got a secret to tell you. You know that computer they fuss over? Well, kid, between you and me, this whole programming thing is a lot simpler than they realize.

What's that? Sure, you can learn. Just get a copy of COMPUTERS FOR KIDS. It's a super book, and it tells you everything you need to know. Huh? You have an Apple? No problem. There's a version just for the Apple. One for the TRS-80 and one for the Atari too, with complete instructions for operating and programming.

The book will take you through everything programmers learn. Its easy to understand and the large type makes it easy to read. You'll find out how to put together a flowchart, and how to get your computer to do what you want it to do. There's a lot to learn, but COMPUTERS FOR KIDS has 12 chapters full of information. You'll even learn how to write your own games and draw pictures that move.

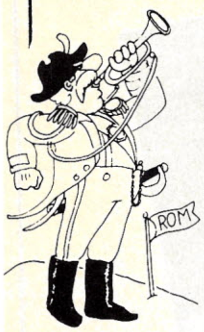
Just so the folks and your teachers won't feel left out, there's a special section for them. It gives detailed lesson ideas and tells them how to fix a lot of the small problems that might pop up. Hey, this book is just right for you. But you don't have to take my word on that. Just listen to what these top educators have to say about it:

Donald T. Piele, Professor of Mathematics at the University of Wisconsin-Parkside says, "COMPUTERS FOR KIDS is the best material available for introducing students to their new computer. It is a perfect tool for teachers who are learning about computers and programming *with* their students. Highly recommended."

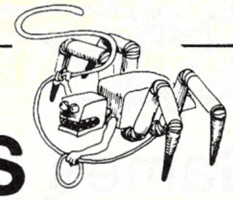
Robert Taylor, Director of the Program in Computing and Education at Teachers College, Columbia University states, "it's a good idea to have a book *for children*."

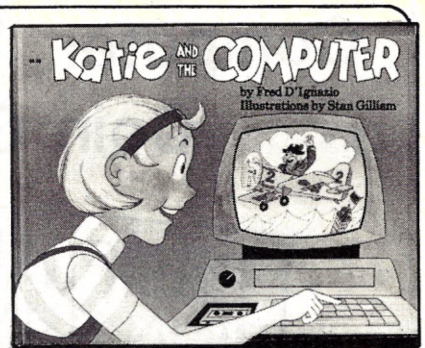
Not bad, huh? Okay, you can let the adults back in the room. Don't forget to tell them COMPUTERS FOR KIDS by Sally Greenwod Larsen costs only $3.95. And tell them you might share it with them, if they're good. TRS-80 (12H); Apple (12G); Atar (12J).

creative computing press



new friends for your child...





Katie and the Computer

Fred D'Ignazio and Stan Gilliam have created a delightful picture book adventure that explains how a computer works to a child. Katie "falls" into the imaginary land of Cybernia inside her Daddy's home computer. Her journey parallels the path of a simple command through the stages of processing in a computer, thus explaining the fundamentals of computer operation to 4 to 10 year olds. Supplemental explanatory information on computers, bytes, hardware and software is contained in the front and back end papers.

Thrill with your chidren as they join the Flower Bytes on a bobsled race to the CPU. Share Katie's excitement as she encounters the multi-legged and mean Bug who lassoes her plane and spins her into a terrifying loop. Laugh at the madcap race she takes with the Flower Painters by bus to the CRT.

"Towards a higher goal, the book teaches the rewards of absorbing the carefully-written word and anticipating the next page with enthusiasm..."

The Leader

"Children might not suspect at first there's a method to all this madness—a lesson about how computers work. It does its job well."

The Charlotte Observer

"...the book is both entertaining and educational."

Infosystems

The book has received wide acclaim and rave reviews. A few comments are:

"Lively cartoon characters guide readers through the inner chamber of the computer."

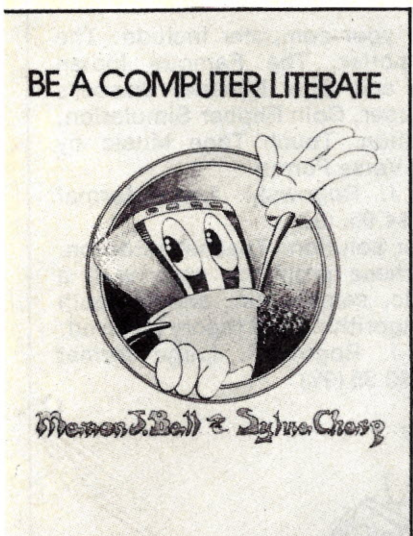
School Library Journal

"...an imaginative and beautifully conceived children's story that introduces two characters—the Colonel and the Bug—who already seem to have been classic children's story book characters for generations."

The Chapel Hill Newspaper

Written by Fred D'Ignazio and illustrated in full color by Stan Gilliam. 42 pages, casebound, $6.95. (12A)

A t-shirt with the Program Bug is available in a deep purple design on a beige shirt. Adult size S, M, L, XL. Children's size S, M, L. $5.00.



Be a Computer Literate

Used as a text in many schools, his informative, full color book is .n ideal first introduction to the vorld of computers for children .ged 10 to 16. The book is divided nto eight chapters:

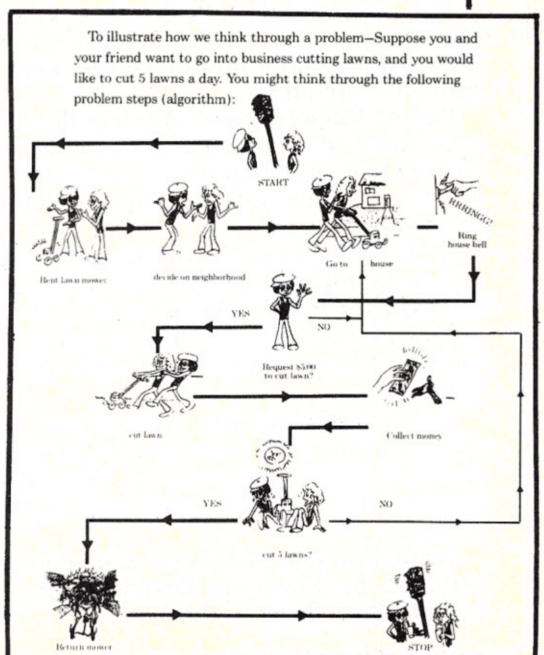
1 Introduction
2 What are computers
3 Kinds of computers
4 What goes on inside computers
5 Communicating with the computer
6 Language of the computer
7 How to write a simple program
8 How computers work for us.

The full color drawings, diagrams and photos found on every page of these chapters, coupled with the large type, make the book easy to read and understand.

The book contains brief explanations of how computers are used in over sixteen different fields, from medicine to law enforcement, art to business. transportation to education.

The simple glossary provided will help familiarize beginners with essential computer terminology.

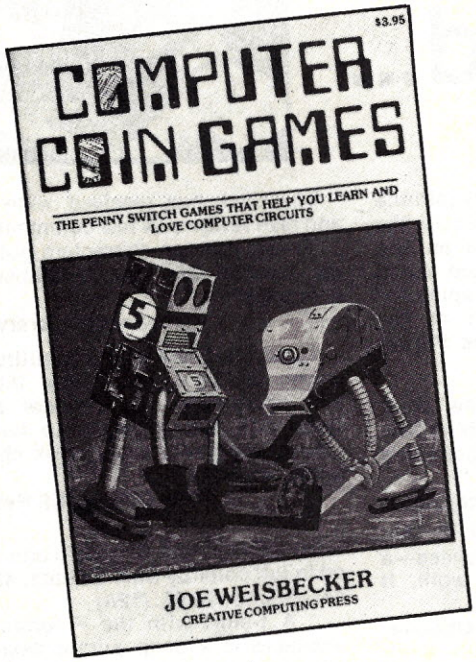
Written by Marion J. Ball and Sylvia Charp. Large format, paperbound, 66 pages, $3.95 (6H)



To order, use bound-in card in back of magazine.

Computer

Coin Games



Learning how computer circuitry works can actually be fun. All you have to do is slide around a few pennies. **Computer Coin Games** presents a series of interesting games with full size playing boards that trace the paths of electronic signals through various simple computer circuits.

Beginning with the "basic penny switch flip flop" the games build in difficulty until the reader is creating intricate networks. Why binary math is used in computers and how it works, how the computer counts, adds, subtracts, uses a number base, and handles letters and words, are all explained in the book.

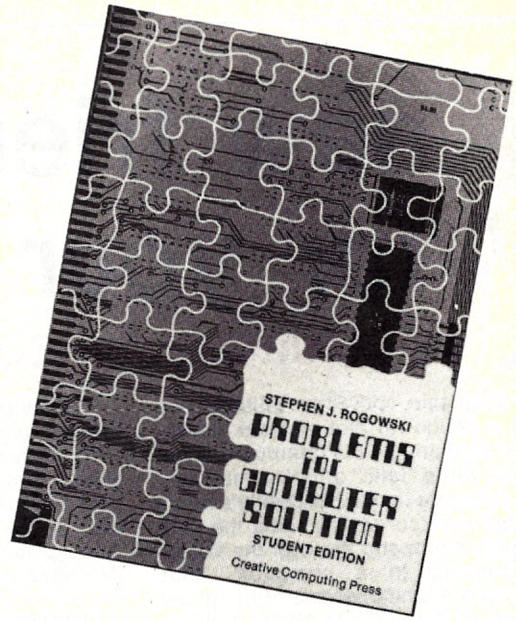
Play "Tic Tac Toe," "Guess a Word," "Create a Pattern" and "Escape the Network." This book is an ideal introduction to the complicated concepts of computer circuitry.

Games Magazine said "Whether or not you have any experience with computer technology you'll be both amazed and delighted by the simplicity of the format and the complexity of the play. All you need is some common cents."

Dr. Dobb's Journal says **"Computer Coin Games** is a simple approach to a complicated concept...**Computer Coin Games** is liberally sprinkled with clever illustrations and diagrams, and provides a relatively painless route to an understanding of how computer circuits function."

The **Association For Educational Data Systems** reports, "An ideal introduction to the complicated concepts of computer circuitry, **Computer Coin Games** features outstanding illustrations by Sunstone Graphics."

Written by Joe Weisbecker and enhanced with great cartoons by Sunstone Graphics. 96 pages, paperbound, $3.95. (10R)



Problems for Computer Solution

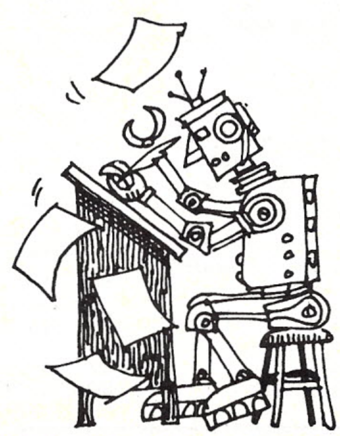
Ninety intriguing and fascinating problems, each thoroughly discussed and referenced, make an excellent source of exercises in research and preliminary investigation. Eleven types of problems are provided in the following areas: arithmetic, algebra, geometry, trigonometry, number theory, probability, statistics, calculus and science. Three classic unsolved problems and seven appendices are also included.

Problems for you and your computer include: The Faulty Speedometer Spotter, The Famous Indian Problem, The Monkey and The Banana, Pascal's Triangle, Perpetual Calender, Coin Flipper Simulation, Einstein's Energy Equation, Touch Tone Music by Computer and Computer Verse Forms

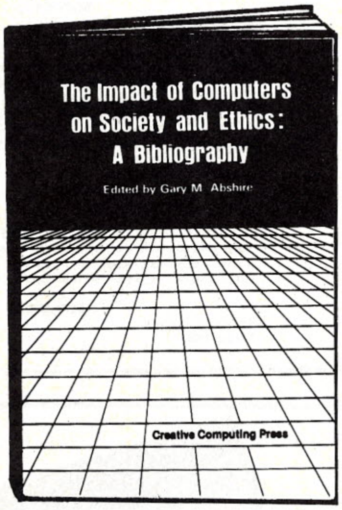
Written by Stephen J. Rogowski. Large format paperbound, 106 pages, $4.95. (9Z)

Problems for Computer Solution: Teacher's Edition, contains solutions to these problems each with a complete listing in Basic, sample run, and in-depth analysis explaining the algorithms and theory involved.

Written by Stephen J. Rogowski. Large format paperbound, 280 pages, $9.95 (9Y)



The Impact of Computers on Society and Ethics: A Bibliography

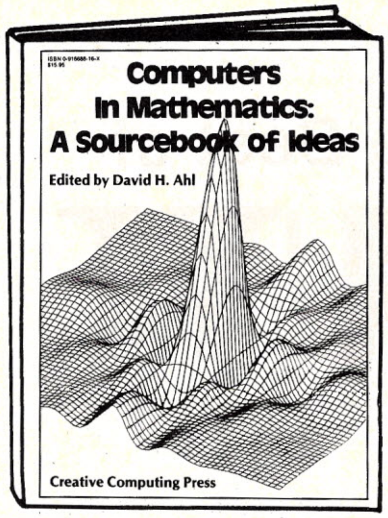


Where is the computer leading us? Is it a menace or a messiah? What are its benefits? What are the risks? What is needed to manage the computer for society's greatest good? Will we become masters or slaves of the evolving computer technology?

This bibliography was created to help answer questions like these. The works cited can provide the range of facts and opinion necessary to your understanding of the role of the computer.

This is a bibliography of works dealing with the ways in which computers are being used in our society, the beneficial changes that are taking place in our lives as a result of computer technology, the social and ethical problems intensified by the improper use of computers, the dangers of computerized society, the safeguards and defenses against those dangers, the attempts to indicate what computerized direction the future will take, and the responsibilities of computer professionals. It contains 1920 alphabetical entries of books, magazine articles, news items, scholarly papers and other works dealing with the impact of computers on society and ethics. Covers 1948 through 1979.

Compiled by Gary M. Abshire. Hardbound, 128 pages. $17.95. (12E).



Computers in Mathematics: A Sourcebook of Ideas

Here is a huge sourcebook of ideas for using computers in mathematics instruction. There are sections on:

*Thinking Strategies and How to Solve Problems
*How to Buy a Microcomputer System
*Art, Graphics, and Mathematics
*Computer Assisted Instruction
*Computer Simulations
*Programming Style
*Probability
*Magic Squares and much more.

One section presents over 250 problems, puzzles and programming ideas, more than are found in most "problem collection" books.

Pragmatic, ready to use, classroom tested ideas are presented for everything from the most basic introduction to binary numbers to advanced techniques like multiple regression analysis and differential equations. Every item discussed has a complete explanation including flowcharts, programs, and sample runs.

The book includes many activities that don't require a computer. And if you're considering expanding your computer facilities you'll find a section on how to select a computer complete with an invaluable microcomputer comparison chart.

Although much of the material has appeared in Creative Computing, many of those back issues are no longer available. Consequently this book meets the demand of making available that popular information.

Edited by David Ahl. Large format paperbound, 224 pages, $15.95. (12D)

creative computing press

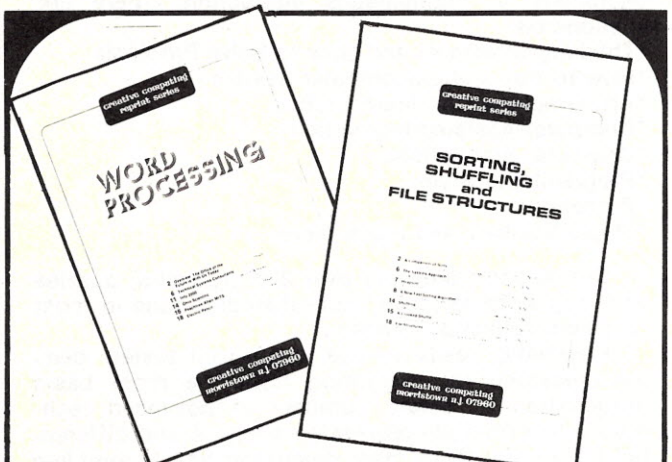
To order, use bound-in card in back of magazine.

The Best of BYTE

This blockbuster of a book contains the majority of material from the first 12 issues of Byte magazine. It is crammed full of how-to articles on everything from TV displays to joysticks to cassette interfaces and computer kits. Also full of software and applications from on-line debuggers to games to a complete small business accounting system. Much more. All of these Byte issues are now out of print so this is the only source of this vital material.

Edited by David Ahl and Carl Helmers. Large format paperbound, 386 pages, $11.95 (6F).

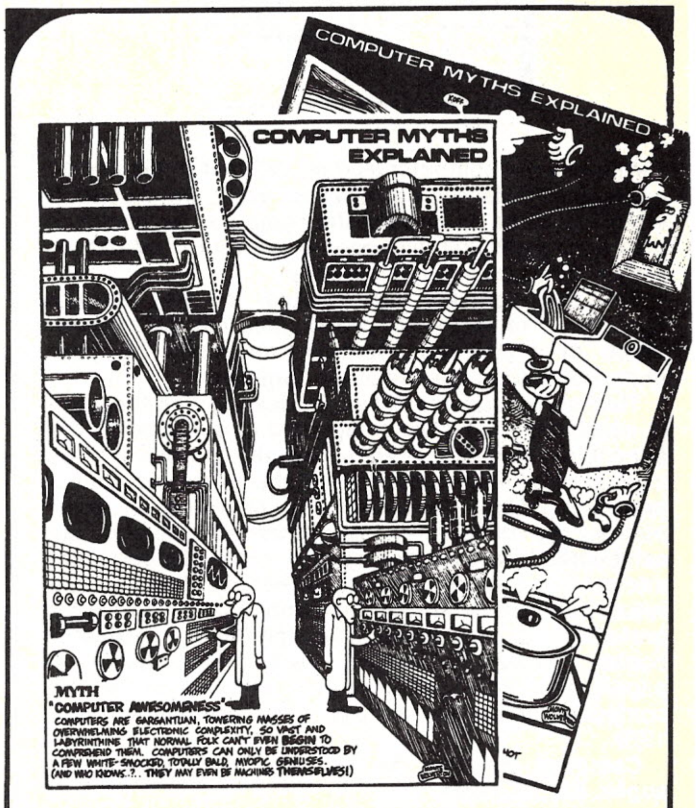
Table of Contents



Word Processing Systems discusses mini- and micro-based systems-- how they work and what they do. It also includes an in-depth desc ription of five representative systems including examples of how they work, sample output and information about availability and prices. Recommended by Free Enterprise and the Journal of Systems Managements.

Sorting and Shuffling has an in-depth discussion of five sorting techniques (bubble, heapsort, Shell-Metzner, delayed replacement, and Woodrum). Also discusses file structures and shuffling techniques. This booklet is a vital necessity if you're doing any programming of your own. Most textbooks either ignore or gloss over these techniques.

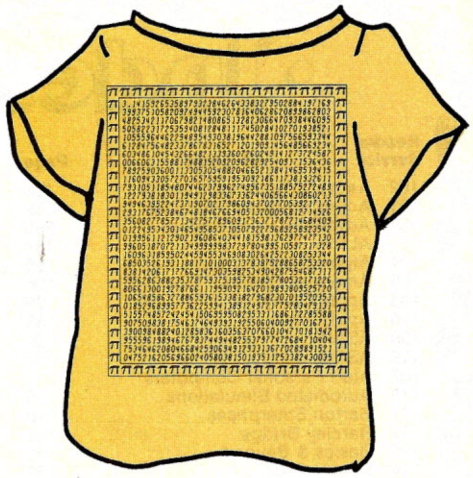
Twenty page reprints of two of Creative Computing's most popular subjects. Each compilation of articles presents an in-depth treatment of its subject. $.50 each.



Set of 8 computer Myths Explained by Monte Wolverton. On heavy stock, large 12x17" size, suitable for framing, dressing up that drab line printer or file cabinet. $3.00

Creative Computing-- Albert Einstein in black on a red denim-look shirt with red neckband and cuffs.

Creative's own outrageous **Bionic Toad** in dark blue on a light blue shirt for kids and adults.



Plotter display of **Pi to 625 Places** in dark brown on a tan shirt.

Give Your Tie a Rest.

I'd rather be playing spacewar-- black with white spaceships and lettering.

T-shirts available in adult sizes S, M, L, XL; and in children's sizes (Bionic Toad, Spacewar and Program Bug) S, M, L. Made in USA. $5.00

Computer Bum-- black design by cartoonist Monte Wolverton on gray denim-look shirt with black neckband and cuffs.

The **Program Bug** that terrorized Cybernia in Katie and the Computer is back on this beige t-shirt with purple design. You can share the little monster with your favorite kid.

Roll down the block with this little black **Robot Rabbit** (on a bright orange t-shirt) on your back and you can intimidate every carrot, radish or cuke in your way.

To order, use bound-in card in back of magazine.

Index to Advertisers

*Write Advertiser Directly

Puzzle Answers

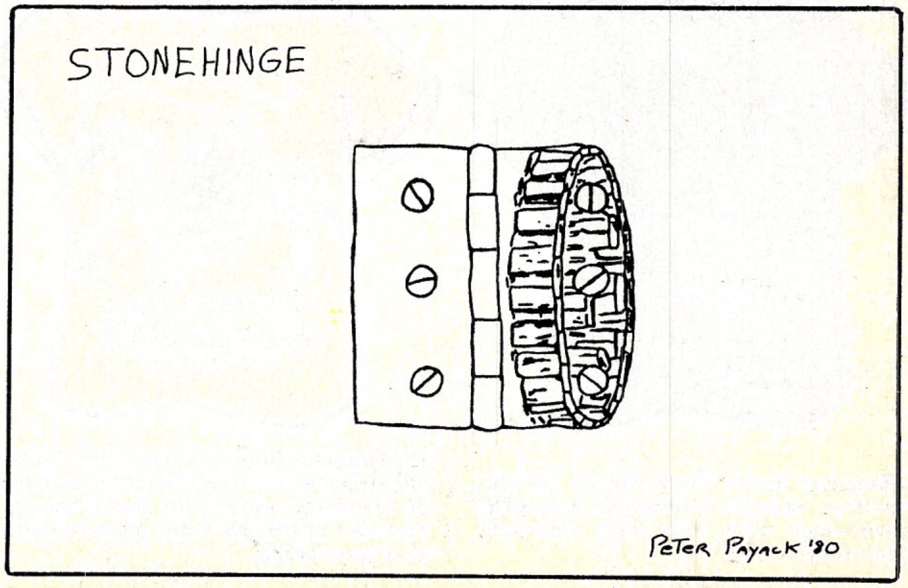
The Time Out Puzzle:

Change The Word Puzzles: FLOP, FLAP, FLAT, FIAT, FIST, FISH, DISH, DISK.

The Over-Polite Guests: To obtain the answer, all that is needed is to find the number of permutations of seven objects (7 x 6 x 5 x 4 x 3 x 2 x 1 = 5040). It would take, therefore 5040 days, or nearly fourteen years, to exhaust the possible positions.

The "Five and Five" Puzzle: For the sake of brevity, we will distinguish the red and black counters by the letters **r** and **b** respectively. They will then stand at the outset as under:

	b	r	b	r	b	r	b	r	b	r	.	.
Position after 1st move;	b			r	b	r	b	r	b	r	r	b
" " 2nd "	b	b	r	r	b	r			b	r	r	b
" " 3rd "	b	b	r			r	r	b	b	r	r	b
" " 4th "	b	b	r	r	r	r	r	b	b			b
" " 5th "	.	.	r	r	r	r	r	b	b	b	b	b



STOP PLAYING GAMES AND GET DOWN TO BUSINESS

Corvus Transforms the Personal Computer into a Powerful Business Tool.

In business, professional offices, and schools throughout the world, thousands of Corvus intelligent peripherals bring mass storage, increased speed, and multi-user capability to a variety of microcomputers. Current applications include accounts receivable and payable, medical records, mailing lists, inventories, word processing, insurance, mathematics and science, and other large and complex files.

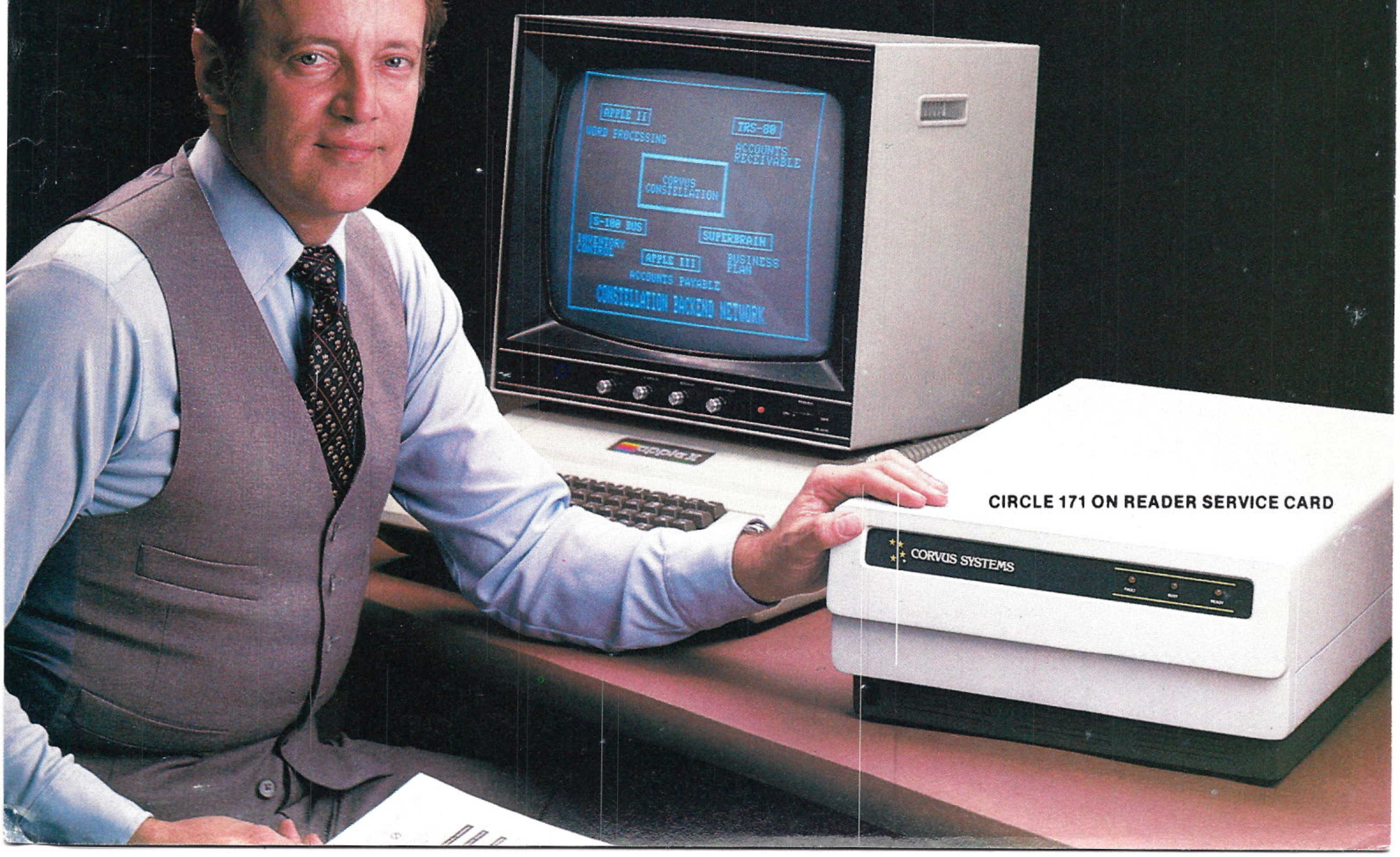
Corvus proven Winchester disk technology provides 10 to 80 million bytes of capacity, fully compatible with your current operating system. This is up to 500 times the capacity of a floppy disk.

The Corvus CONSTELLATION links up to 64 computers in a state-of-the-art multi-processor network. It provides shared mass storage, pipes for inter-computer communication, and system spooling for sharing of peripherals such as printers. Performance far exceeds that of larger and more expensive networks.

Backup data protection and archival storage are provided by the Corvus MIRROR (Patent Pending), a low-cost backup using standard video cassette recorders.

Contact your local Corvus dealer for the full story about these innovative new products.

CORVUS SYSTEMS
2029 O'Toole Avenue
San Jose, California 95131
(408) 946-7700/TWX 910-338-0226



Educator, Entertainer, Accountant.

Your Challenger Personal Computer.

Through the miracle of modern technology, a complete computer as powerful as the multimillion dollar room-sized computers of a few years ago can be put in a package the size of a typewriter and sells for as little as a color television set!

Through its years of microcomputer experience, Ohio Scientific has effectively channeled this tremendous computer power into a "friendly" computer with hundreds of personal uses, via a huge software library of programs for a broad range of personal, home, educational and business use.

This available software allows you to use and enjoy your computer without becoming an expert. The Challenger, however, is a powerful, general purpose computer which can be programmed in several languages by those who choose to.

Here are just a few of the popular uses of an Ohio Scientific Challenger Computer:

Education

The personal computer is the ultimate educational aid because it can entertain while it educates. Software available ranges from enhancing your children's basic math, reading and spelling ability, through tutoring high school and college subjects, to teaching the fundamentals of computers and computer programming.

Entertainment

Many of the Challenger's games educate while they entertain, from cartoons for preschoolers to games which sharpen mathematical and logical abilities. But, entertainment doesn't stop here. The Challenger's graphics capabilities and fast operation allow it to display action games with much more detail than the best video games, providing spectacular action in games such as Invaders, Space Wars, Tiger Tank and more! All popular sports such as golf, baseball and bowling are available as simulated computer games as well as many conventional games such as chess where the computer plays the role of a formidable opponent.

Accounting

Your Challenger computer can keep track of your checkbook, savings account, loans, expenses, monitor your calorie intake and your biorythms.

If you are involved in a business, you can use it to do word processing; accounting, inventory control, order processing, customer lists, client records, mailing labels and planning.

And more:

This may seem like a lot of uses, but it's only the tip of the iceberg for a general purpose computer. For example, your Challenger can be expanded to control lights and appliances, manage your energy usage and monitor for fire and break-ins. Furthermore, it can communicate with you, with other computers and the new personal computer information services over the telephone.

In fact, the uses of general purpose, personalized computers are expanding daily as more and more people discover the tremendous capabilities of these new technological wonders.

Ohio Scientific offers you four personalized computer systems starting at just $479.

For a free catalog and the name of the dealer nearest you, call 1-800-321-6850 toll free.

OHIO SCIENTIFIC
1333 SOUTH CHILLICOTHE ROAD
AURORA, OH 44202 • (216) 831-5600

CIRCLE 172 ON READER SERVICE CARD